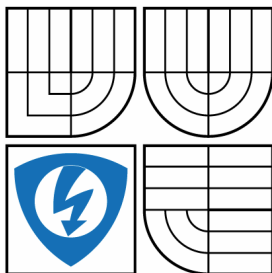


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

## STATICKÝ MODEL SCÉNY

STATIC MODEL OF SCENE

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

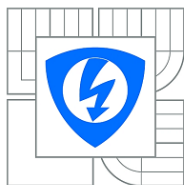
Bc. JAN SIKORA

### VEDOUcí PRÁCE

SUPERVISOR

Ing. MILOSLAV RICHTER, Ph.D.

BRNO 2013



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí techniky

## Diplomová práce

magisterský navazující studijní obor  
Kybernetika, automatizace a měření

**Student:** Bc. Jan Sikora  
**Ročník:** 2

**ID:** 115115  
**Akademický rok:** 2012/2013

### NÁZEV TÉMATU:

**Statický model scény**

### POKYNY PRO VYPRACOVÁNÍ:

Úkolem je vytvořit aplikaci, která na základě snímků pořízených ve větších časových odstupech vytvoří snímek pozadí (neboli odstraní ze snímku pohyblivé objekty).

Provedte popis typů scén, a možných využití této aplikace. Vytvořte databázi snímků pro různé typy scén. Naprogramujte algoritmy pro detekci objektů ve snímcích a jejich lokalizaci na snímcích pořízených v různých okamžicích. Navrhněte způsoby jak určit, které objekty patří k pozadí. Ze snímků extrahujte pozadí a uložte výsledky ve vhodném formátu.

Zhodnotte zvolené řešení a dosažené výsledky.

### DOPORUČENÁ LITERATURA:

Žára J., Beneš B., Sochor J., Felkel P.: Moderní počítačová grafika, Computer Press, 1998, ISBN 80-251-0454-0

Hlaváč V., Šonka M.: Počítačové vidění, Grada, Praha 1992, ISBN 80-85424-67-3

Faugeras O.: Three-Dimensional Computer Vision, The MIT Press 1993

Kraus K.: Photogrammetrie 1 und 2, Ummler / Bonn, 1996

**Termín zadání:** 11.2.2013

**Termín odevzdání:** 20.5.2013

**Vedoucí práce:** Ing. Miloslav Richter, Ph.D.

**Konzultanti diplomové práce:**

**doc. Ing. Václav Jirsík, CSc.**

*Předseda oborové rady*

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **Abstrakt**

Práce se zabývá různými metodami detekce pozadí a s tím související detekcí pohybu ve scéně. Je postupováno od nejjednodušších metod k složitějším. U každé jsou zhodnoceny možnosti použití, její nedostatky. V úvodu jsou rozebrány různé typy scén podle stylu pozadí a popředí např. podle rychlosti pohybu objektů nebo výskytu pohybu v pozadí. Je navrženo několik obecných nebo specifických vylepšení pro získání kvalitnějšího pozadí i použitím jednoduché metody. Další část práce řeší reálnou situaci chvějící kamery. Jsou vyzkoušeny dvě základní metody na optickou stabilizaci obrazu. První je sesouhlasení pomocí vyhledávání šablony v obrázku. Druhá pomocí významných (rohových) bodů. Obě metody jsou podrobně zkoušeny a je hledán nejlepší způsob sesouhlasení po sobě následujících snímků. Kromě chvějící se kamery se práce zabývá i otáčivým pohybem kamery a teoretickým návrhem řešení detekce pozadí z kamery umístěné na jedoucím autě. Součástí práce je vytvoření databáze různých typů scén.

## **Klíčová slova**

**Detekce pozadí, mixture of Gaussian, registrace obrazů, detekce rohových bodů, porovnávání šablon**

## **Abstract**

This thesis deal with various methods of background detection and with it related motion detection in a scene. It's progressing from simplest methods to more complex. For every one are reviewed the possibilities of using and her drawbacks. In introduction are described various types of scenes according to background and foreground type e.g. according to movement objects speed or presence of movement in background. Is proposed several common or specific improvements for obtaining better background even by using simple method. Next part of work solve real situation of shaking camera. There are tested two basic methods for optical stabilization. The first is registration of images by template matching. Alternative method used interest points (corners). Both methods are closely examine and is sought best way to match following pictures. Except shaking of camera this work deal with rotating camera and in theory solve detection background from cameras placed on ridden car. Part of work is creation database of different types scenes

## **Keywords**

**Background detection, mixture of Gaussian, image registration, corner detection, template matching**

### **Bibliografická citace:**

**SIKORA, J. *Statický model scény*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 72s. Vedoucí diplomové práce byl Ing. Miloslav Richter, Ph.D.**

## **Prohlášení**

**„Prohlašuji, že svou diplomovou práci na téma Statický model scény jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.**

**Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.**

V Brně dne: **17. května 2013**

Jan Sikora  
podpis autora

## **Poděkování**

**Děkuji vedoucímu diplomové práce Ing. Miloslavu Richterovi, Ph.D. za pedagogickou a odbornou pomoc a další užitečné rady při zpracování mé diplomové práce.**

**V Brně dne: 17. května 2013**

Jan Sikora  
podpis autora

# Obsah

1 Úvod.....	10
2 Teoretický úvod.....	11
2.1 Úvod.....	11
2.2 Využití.....	11
2.3 Možné typy scén.....	11
3 Algoritmy detekce pozadí.....	12
3.1 Rozdílové metody (Background Subtraction Methods, Basic BGS).....	13
3.2 Running Gaussian average.....	14
3.3 Temporal median filter.....	15
3.4 Mixture of Gaussian.....	16
3.5 Kernel density estimation.....	19
3.6 Jiné způsoby detekce pozadí.....	21
3.7 Různá obecná zlepšení .....	21
3.8 Hodnocení algoritmů.....	22
3.9 Použití barevného videa.....	23
3.9.1 Barevné modely.....	23
3.9.2 Měření podobnosti barev.....	24
4 Pohybující se kamera.....	25
4.1 Sesouhlasení snímků.....	25
4.2 Metoda využívající korelace.....	25
4.2.1 Normalizovaná korelace.....	27
4.3 Metoda využívající významných bodů.....	28
4.3.1 Detektory významných bodů.....	28
4.3.2 Moravcův operátor.....	28
4.3.3 Harrisův detektor.....	28
4.3.4 Nalezení odpovídajících si významných bodů.....	31
4.3.5 RANSAC algoritmus (Random Sample Consensus).....	32
4.4 Detekce pozadí při použití otáčející se kamery.....	32
4.5 Detekce pozadí z libovolně se pohybující kamery.....	33
5 Popis implementace některých algoritmů a výsledky testů.....	34

5.1 Detekce pozadí.....	34
5.1.1 Obecně ke každé metodě.....	34
5.2 Post-processing popředí.....	36
5.2.1 Lokálně proměnné prahování.....	36
5.2.2 Zaplnění velkých děr.....	37
5.3 Mixture of Gaussian.....	38
5.4 Detekce pohybu v barevném videu.....	42
5.5 Použití významných bodů.....	46
5.5.1 Moravcův operátor.....	46
5.5.2 Harrisův detektor.....	46
5.5.3 Metody sesouhlasení z významných bodů.....	47
5.6 Korelační metody.....	49
5.6.1 Optický stabilizátor pomocí korelace.....	49
5.6.2 Normalizovaná korelace a automatické vyhledání šablony.....	50
5.6.3 Otáčející se kamera.....	52
6 Závěr.....	54



## Seznam obrázků

Obr. 1 Vliv nastavení prahu při prahování rozdílového snímku (bílé pixely – pohyb).....	13
Obr. 2 Vliv morfologických operací.....	14
Obr. 3 „duchové“ v krátkém bufferu.....	16
Obr. 4 Rozložení Gaussianů při adaptaci na dvě různá pozadí.....	17
Obr. 5 Vysvětlení k aproximaci pravděpodobnosti $P(k X, \Phi)$ [10].....	18
Obr. 6 Geometrická reprezentace barevného modelu HSV [13].....	23
Obr. 7 Ukázka dobře použitých morfologických operací při získání pozadí.....	35
Obr. 8 Lokálně proměnné prahování.....	37
Obr. 9 MoG model, zleva aktuální obrázek, popředí a pozadí.....	41
Obr. 10 Vliv periodičnosti HSV modelu na rozdílový obrázek.....	44
Obr. 11 Výchozí obrázek k ukázce vlivu periodičnosti HSV modelu.....	44
Obr. 12 Odstranění periodičnosti HSV modelu.....	45
Obr. 13 Harrisův rohový detektor.....	47
Obr. 14 Korespondence rohových bodů při malé minimální blízkosti bodů .....	48
Obr. 15 Korespondence rohových bodů při větší minimální blízkosti bodů.....	48
Obr. 16 Ukázka optického stabilizátoru na základě korelace.....	49
Obr. 17 Špatná stabilizace vlivem nevhodně vybraného objektu.....	50
Obr. 18 Korelační obrázek na první pohled vhodné šablony.....	51
Obr. 19 Výsledek automatického vyhledání šablony.....	52

# 1 ÚVOD

Tato práce se zabývá extrakcí pozadí z videa, nebo sekvence snímků. Pozadí je běžně chápáno jako velké statické objekty ve scéně. Pozadí však většinou není dokonale statické, ale pomalu se mění např. vlivem různé denní doby. Do pozadí jsou také obvykle řazeny např. stromy nebo vodní plochy. Zde už nelze, hlavně za větrného počasí mluvit o statických objektech. Jiný příklad proměnného pozadí jsou semaforey na křižovatkách nebo jakákoliv periodicky blikající světla. Práce rozděluje typy pozadí do různých scén.

Dále jsou teoreticky popsány některé algoritmy pro získání pozadí ve scéně. Díky existenci různých typů scén je možné pro některé z nich použít i ten nejjednodušší popsáný algoritmus. Pro jiné scény naopak nedává uspokojivé výsledky a je nutné nasadit složitější metodu. Každá má své výhody a nevýhody. Např. obvykle čím přesnější výsledky metoda vykazuje, tím déle její výpočet trvá. Jako v některých jiných úlohách počítačového vidění je taky u této úlohy použití konkrétního algoritmu velmi aplikačně závislé.

Část práce je věnována zpracování výsledků s cílem zvýšit jejich kvalitu bez ohledu na to jaký algoritmus je použit. K tomu patří i ošetření otřesů kamery. Jsou navrženy 2 metody pro registraci snímků. První je založena na korelacích a druhá na významných bodech. Kromě parazitních otřesů kamery je uvažována i rotující kamera a kamera připevněná na automobilu. Protože přesné sesouhlasení snímků je pro následnou detekci pohybu velmi důležité jsou popsány algoritmy podrobněji prozkoumány a otestovány.

Některé popsány algoritmy byly naprogramovány, otestovány a zhodnoceny. Praktická část práce uvádí dosažené výsledky a řešení problémů, které při tom nastaly. Součástí práce bylo také vytvoření databáze různých typů scén, na kterých by mohli být tyto algoritmy testovány. Scény byly vybrány na základě konkrétních možností využití detekce pozadí (a s tím související detekcí pohybu), např. scéna „dálnice“.

Práce nebude popisovat základní postupy zpracování obrazu, jako např. morfologické operace.

## 2 TEORETICKÝ ÚVOD

### 2.1 Úvod

Statické model scény je obvykle získán z nepřetržitého videa, kde se pohybující se objekty (popředí) vyskytují na snímku současně se statickými (pozadí). Rozdíl mezi nimi je v existenci nebo poloze na následujícím snímku. V některých případech, je lepší použít místo videa sekvenci snímků ve větších odstupech., klidně i náhodných. To lze použít při velmi pomalu se pohybujících objektech nebo při krátkodobě stálém, ale dlouhodobě se měnícím popředí (auta na parkovišti, pokud chceme získat snímek prázdného parkoviště). Získaný model scény se dále zpracovává. Obvykle se odečte aktuální snímek. Na snímku zůstanou pohyblivé objekty a výsledek je většinou prahován. V původním obrázku pak může být zvýrazněn pohybující se objekt, např. při trasování.

### 2.2 Využití

Získané pozadí samo o sobě příliš využití nemá, snad jen pokud by bylo potřeba fotku prázdné haly nebo jiného místa (do tisku nebo prospektu) a ono halu by nebylo možné uzavřít pro veřejnost.

Hlavní místo využití je při detekci pohybu ve scéně. Jak je psáno výše při detekci pohybu je od aktuálního snímku odečteno pozadí a tím zvýrazněny pohybující se objekty. To lze použít hlídání střeženého objektu, pohybu aut ve městě, na parkovišti nebo osob na letišti. Případně při ovládání zařízení pomocí gest, tvorby mapy prostředí u pohyblivých robotů,...[1] V těchto konkrétních případech byla využita detekce pozadí a následná detekce pohybu: zmíněný dopravní monitoring, jako sledování hustoty dopravy, dohledová kamera na ulici [1], detekce nebezpečných situací při shromažďování velkého davu osob [2], sledování pohybu mraků na obloze [3]. Dále bylo toto použito při „automatických“ parkovacích hodinách, kde byly detekovány jednotlivá auta pomocí statického pozadí ulice. Jiné využití by mohlo být při optimalizaci složitých algoritmů ve kterých by se zpracovávaly pouze oblasti kde došlo k pohybu (algoritmus detekce by musel být dostatečně nenáročný, aby k nějaké optimalizaci vůbec došlo).

Programy vzniklé v souvislosti s optickou stabilizací kamery mají, kromě hlavní funkce i spoustu dalších využití, které už ale nejsou předmětem práce.

### 2.3 Možné typy scén

V ideálně laboratorním použití je testována tato scéna. Neměnné pozadí (z hlediska přesunu objektů i jejich jasu nebo barvy nebo proměnnosti pozadí), jeden nebo více navzájem se na obrázku oddělitelných pohybujících se objektů a statická kamera.

Taková to scéna se v reálném použití vyskytuje snad jen při kontrole součástek na pásu tovární výrobě. I tam ale nemusí být všechny podmínky splněny, navíc bude vždy v obraze působit šum.

V praktickém použití se objevuje ve venkovních prostorách: postupná i náhlá změna jasu nepohyblivého pozadí vlivem slunce, mraků a obecně počasí. V halách může dojít k náhlému zvýšení jasu díky rožnutí světla. Barva statických objektů se obvykle nemění, oproti tomu má často pohybující se objekt konstantní barvu na celé jeho ploše, což vede detekci pouze okrajů např. velké bílé dodávky. Ve venkovních prostorech je dále častý problém s periodicky se pohybujícím pozadím, které přesto je nutné určit jako pozadí, např. větve stromů, vlnící se vodní hladina, ... Dalším problémem jsou pohyblivé stíny. [4] Může dojít ke dlouhodobým změnám pozadí, zastavením nebo odjetím auta na ulici, zasněžením ulice, ... Pohyblivé objekty se mohou pohybovat velmi malou rychlostí. V případě aktualizovaného pozadí pak s ním splynou. Kromě toho mohou být z části průhledné (skla na autech). Dalším praktickým problémem je pohyb kamery, ať už záměrný, v případě kamery na jedoucím autě, nebo nechtěný.

Vyskytují se tyto typy scén:

- chodba: nepohyblivé pozadí, neměnné pozadí, pohyb osob, blízké k ideálnímu
- hala: nepohyblivé pozadí, možnost proměnného pozadí, vliv Slunce přes okna
- ulice: pohyblivé pozadí (větve stromů)
- křižovatka: v oblasti zájmu pouze statické pozadí, vliv denní doby
- parkoviště: jako na ulici, jen jiné použití výsledků
- dálnice: rychlý pohyb objektů, běžně neměnné pozadí
- v přírodě: množství pohybujícího se pozadí, velmi obtížné
- metro: dvě proměnná pozadí (nástupiště, drážní souprava)

Snad kromě scény chodba a dálnice, nebo silničního tunelu je vždy možná dlouhodobá změna pozadí. Právě toto nastavení času, od kterého se objekt považuje za součást pozadí, nebo zda vůbec je za pozadí považován je aplikačně závislé.

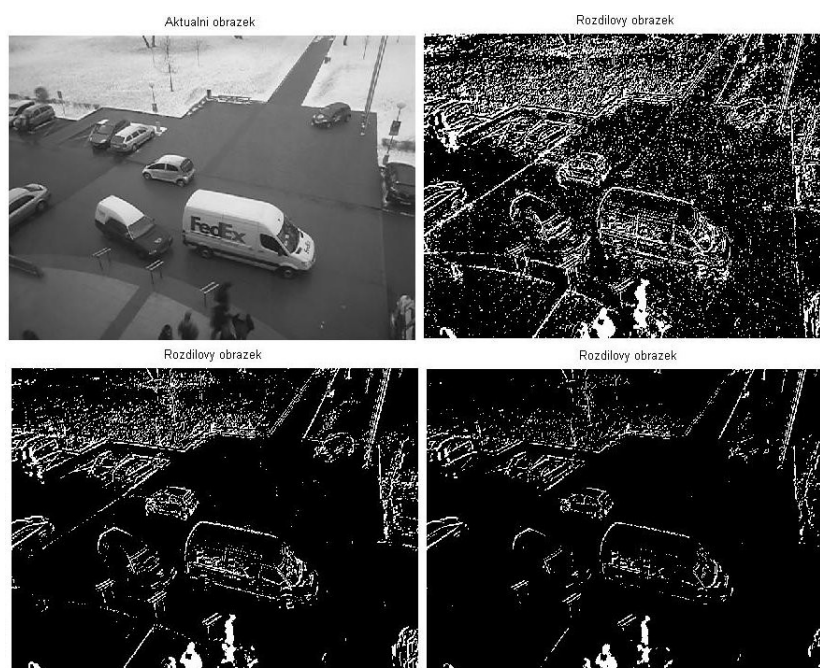
Z tohoto souhrnu je vidět, že venkovní scény jsou problematičtější, a že nelze vytvořit univerzální postup. Ovšem některé složitější algoritmy umí potlačit více těchto problémů záraz (např. Mixture of Gaussian).

### 3 ALGORITMY DETEKCE POZADÍ

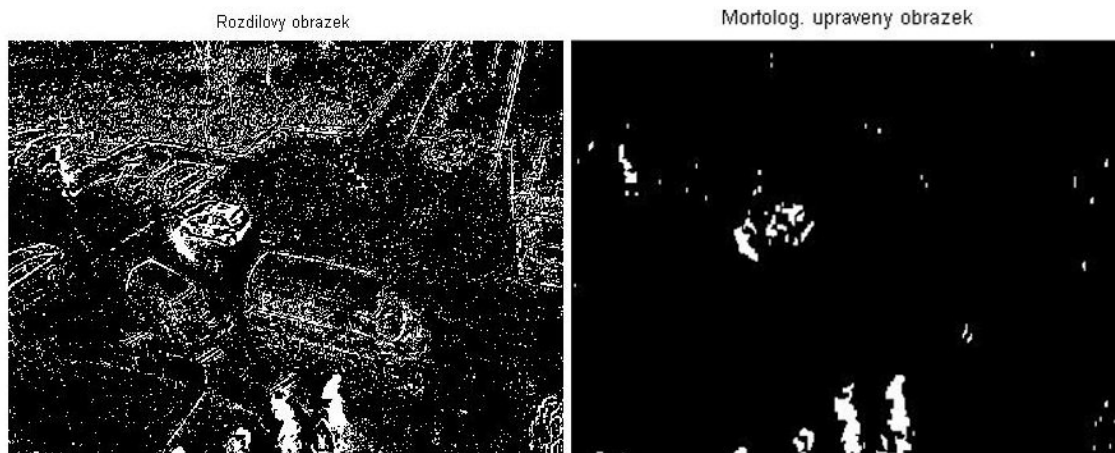
Následuje popis některých používaných algoritmů. Nejedná se o naprostý výčet, jen o základní typy, ke kterým se pak většinou přidávají různá vylepšení na odstranění konkrétních problémů, například zmíněných stínů nebo duchů (uměle vzniklých útvarů, které ale nepatří k reálnému objektu [5], automatické prahování výsledků [6], zaplnění „děravých“ pohyblivých objektů [7], ...

### 3.1 Rozdílové metody (Background Subtraction Methods, Basic BGS)

Jde o klasickou metodu vycházející z detekce pohybu. Od aktuálního snímku je odečten předcházející. Tím se vytvoří rozdílový snímek, na kterém jsou pohybující se objekty vyznačeny světlejší barvu než statické pozadí. To by mělo být ideálně černé. Výsledný rozdílový snímek je prahován zvoleným prahem. Získá se binární snímek kde bílé pixely označují pohybující se objekty. Pokud se na snímku nevyskytují bílé pixely může být aktuální snímek označen za pozadí. Taková situace např. v letištní hale téměř nenastane. Proto lze snímek dělit na oblasti, případně rovnou testovat na pohyb jednotlivé pixely. Bylo ověřeno, že kvalita výsledků velmi závisí na nastavení prahu, viz Obr. 1. Z obrázku je vidět, že při libovolném prahu se vždy vyskytuje množství šumu, přičemž už jde o obrázky kde je šum filtrován průměrováním z 5 po sobě jdoucích snímků. Mezi možnosti odstranění šumu patří zmíněná filtrace, filtrace nějakým druhem filtrační masky, použití mediánového filtru nebo použití morfologické operace eroze (otevření) s nastaveným strukturním elementem. Použití vhodného elementu je opět klíčovým prvkem pro kvalitu výsledku, ale i pro základní element kříž 3x3 je výsledek mnohem lepší, viz Obr. 2. Např. jediné pohybující se auto ve scéně bylo to bílé zadní. Na ostatních autech byly vyznačeny jen hrany, díky chvění kamery. Tento vedlejší efekt byl pomocí morfologie odstraněn. Ostatní Pro zaplnění děr v objektech objektů bylo po erozi ještě použito morfologické uzavření, element kříž 5x5.



Obr. 1 Vliv nastavení prahu při prahování rozdílového snímku (bílé pixely – pohyb)



**Obr. 2 Vliv morfologických operací**

Aktualizace celkového pozadí v této metodě je řešena klouzavým průměrkováním nebo mediánem z určité posloupnosti snímků pozadí. Pokud na daném pixelu nenastal pohyb může být aktuální pozadí zařazeno do průměru. [4]

Výhody metody: jednoduchost implementace; průhlednost algoritmu, výpočetní nenáročnost

Nevýhody metody: výsledek závisí na nastavení prahu; omezeně, častěji vůbec řeší pohyb v pozadí (stromy ve větru), neřeší problémy s homogenně barevnými oblastmi pokud je použito mediánové aktualizování algoritmu vyžaduje množství paměti pro všechny snímky posloupnosti. [4]; nastavení prahu i strukturních elementů musí být uděláno ručně na konkrétní podmínky, pro jiné podmínky je většinou potřeba jiné nastavení.

Kvalitu lze dále zvýšit. Zlepšení původně předpokládané při použití barevných snímků a nebylo tak vysoké jak bylo očekáváno. Různá jiná zlepšení budou popsány dále. Metoda je vhodná pro scény typu chodba, pokud není kamera nevhodně instalována (lidé přicházejí kolmo ke kameře a vzniká problém s homogenně barevnými oblastmi) nebo typu hala.

## 3.2 Running Gaussian average

V tomto postupu je použita statistická informace nasbíraná z několika předchozích snímků. Hodnoty konkrétního pixelu se v čase mění. Pokud se jedná o pozadí, jeho hodnoty se mění pouze vlivem šumu nebo pomalu se měnících světelných podmínek. Hodnoty pixelu tak vytvářejí Gaussovo rozložení. Ke každému je přiřazena informace o střední hodnotě a směrodatné odchylce tohoto rozložení. Jestli je aktuální hodnota pixelu odpovídá pozadí nebo popředí se testuje jeho vzdáleností hodnoty od střední hodnoty. Pokud je splněno:

$$|I_{akt} - \mu| < k * \sigma, \quad (1)$$

kde  $I_{akt}$  je aktuální pixel/obrázek,  $\mu$  střední hodnota,  $\sigma$  směrodatná odchylka a „k“ volitelná konstanta. Aby nebylo nutné uchovávat dlouhou posloupnost snímků pro

výpočet parametrů rozložení, je použit klouzavý průměr (running average) pro jejich aktualizaci. Ta je prováděna podle vzorců:

$$\mu(t+1) = \alpha * I_{akt}(t) + (1 - \alpha) * \mu(t) \quad (2)$$

$$\sigma^2(t+1) = \alpha * (I_{akt}(t) - \mu(t))^2 + (1 - \alpha) * \sigma^2(t), \quad (3)$$

kde  $\alpha$  je koeficient, kterým se nastaví rychlost aktualizace pozadí.[8]

Výhody: relativní jednoduchost; průhledný algoritmus; méně závislé na nastavení prahu (v podobě konstanty „k“) než u rozdílových metod; vyšší kvalita výsledků, díky použití přirozeně fungujícího Gaussova rozložení; podle nastaveného  $\alpha$  schopné dostatečně rychlého obnovování pozadí při jeho změně;

Nevýhody: výsledek stále závisí na nastavení konstanty „k“; problém nastavení adaptačního parametru  $\alpha$ , pokud se ve scéně vyskytují objekty pohybující se různou rychlostí; neschopnost odstranění pohybu v pozadí, rychle se měnícího (větve stromů).

Zlepšení: Pro zvýšení rychlosti je možné parametry  $\mu$  a  $\sigma$  aktualizovat pouze jednou za několik snímků. [8] Naopak pro zvýšení kvality výsledku nenastavovat parametr  $\alpha$  a proměnnou „k“ na jednu hodnotu pro celý snímek, ale s ohledem rozmístění pozadí a objektů v konkrétní scéně. Např. ve scéně dálnice by na vozovce mohly být nastaveny jinak než mimo dálnici. Tím by se mohla spolehlivě odstranit rychle jedoucí auta a přeci jen částečně odstranit i pohyb v pozadí např. kukuřičných klasů na poli. To vše ale za cenu nutného přizpůsobování dané scéně.

### 3.3 Temporal median filter

K detekci pozadí se používá metoda mediánovou hodnotu z bufferu „n“ snímků. Aktualizace bufferu probíhá odstraněním nejstaršího snímku a přidání snímku aktuálního. Podle různých autorů [podle [8]] může tato metoda dávat dobré nebo alespoň shodné výsledky s předchozí popisovanou metodou. Jsou získány dobré výsledky i když do bufferu není použit každý snímek ale jen třeba každý desátý [8]. Kvalita pozadí závisí na délce bufferu. Pokud je použit příliš krátký, může dávat pozadí s „duchy“ (částmi průhledných postav po pohybujících se lidech; ne ve smyslu popsaném v úvodu kapitoly 3.4) viz. Obr. 3.

Výhody: relativně stabilní, rychlý algoritmus, schopnost práce i s každým x-tým snímkem[8]

Nevýhody: vysoká paměťová náročnost podle délky bufferu (musí se uložit každý snímek celý), se jeho vzrůstající délkou pomalá odezva



**Obr. 3 „duchové“ v krátkém bufferu**

Na prvním obrázku je aktuální snímek, osoba uprostřed haly je v pohybu. Na druhém je kousek za osobou vidět její „duch“. Je použit buffer 10 snímků. Pro třetí obrázek je použito 15 snímků. Duch je viditelně slabší. Buffer proto musí být přizpůsoben konkrétní rychlosti pohybu objektů ve scéně a čím delší buffer tím víc je potřeba paměti.

V rámci mediánového filtru byl vyzkoušen i „Temporary mode filter“, který jako pozadí vracel nejčastější hodnotu. Výsledky odstranění pohybujících se osob byly velmi dobré. Okraje hran objektů pozadí byly ale velmi neostré a zašuměné. To bylo způsobeno použitím nestabilizovaného videa. Při použití optické stabilizace, nebo stativu bude výsledek velmi dobrý. Nevýhodou výběru nejčastějšího prvku je vysoká výpočetní náročnost.

### 3.4 Mixture of Gaussian

Dále popisováno jako MoG (Mixture of Gaussians). Pomocí tohoto algoritmu lze odstranit periodicky se opakující pohyb v pozadí, např. když jeden pixel opakovaně zobrazuje větve, listy a budovu za stromem. Každý pixel tedy mění svůj jas (barvu) do několika hodnot. Každé pozadí je popsáno jedním Gaussovým rozložením („Gaussianem“) (kvůli šumu) s střední hodnotou  $\mu$ , směrodatná odchylka  $\sigma$  a váhou  $w$  (lze přirovnat k výšce křivky rozložení), kterou zohledňuje jak často se dané rozložení objevuje. Na aktuálním snímku  $I_{akt}$  se na každém pixelu pro všechna rozložení (běžně 3-5) zjistí jestli jeho hodnota odpovídá některému z rozložení (v [9] i jinde nazýváno „match“). Pokud je splněna podmínka

$$|X_{akt} - \mu| < 2,5 * \sigma, \quad (4)$$

kde  $X_{akt}$  je aktuální hodnota pixelu, pak je rozložení označeno jako „match“.

Když je takto vybráno správné rozložení, předpokládá se, že se toto rozložení vyskytuje na snímku častěji a tedy může být aktuálním pozadím. Pro toto rozložení se zvýší jeho váha, střední hodnota se posune blíže k  $X_{akt}$  a sníží se jeho směrodatná odchylka, podle vzorců:

$$w(t) = (1 - \alpha) * w(t-1) + \alpha * M(t) \quad (5)$$

$$\mu(t) = (1 - \rho) * \mu(t-1) + \rho * X_{akt}(t) \quad (6)$$

$$\sigma^2(t) = (1 - \rho) * \sigma^2(t-1) + \rho * (X_{akt} - \mu)^T * (X_{akt} - \mu) \quad (7)$$



$$\rho = (\alpha * M(t)) / w$$

kde  $M(t) = 1$  pro „match“ jinak se  $M(t) = 0$  a  $\alpha$  je adaptační koeficient (v [10] doporučen 0,005) U rozložení které nebylo označeno jako match se sníží váha podle podobné rovnice jako (5).

Pokud žádné z rozložení není označeno jako match, jedná se buď o popředí nebo o nový druh pozadí. Pak se vybere rozložení s nejnižší váhou a jeho střední hodnota se nahradí  $X_{akt}$ . Směrodatná odchylka se nastaví na vysokou hodnotu. Váha se sníží na malou hodnotu. V [10] se doporučují tyto hodnoty (jen pro počet rozložení = 3):

$$\mu (init) = X_{akt}$$

$$\sigma^2 (init) = 30$$

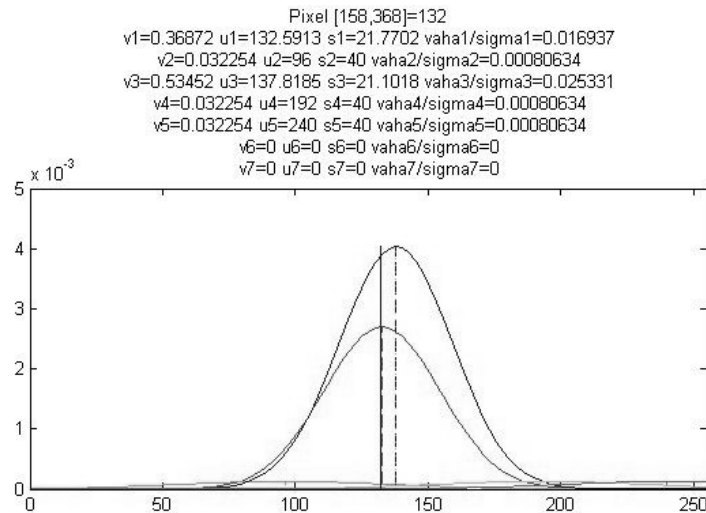
$$w(init) = 0,05$$

Tímto bylo určeno popředí. Po několika snímcích má nejčastěji se objevující rozložení vysoko váhu a malou směrodatnou odchylku. Podle toho je vybráno správné pozadí. Všechna rozložení jsou seřazena podle hodnoty  $w/\sigma$ . Nejpravděpodobnější rozložení je tak první v řadě. To správné je vybráno podle rovnice:

$$B = \arg \min_b \left( \sum_{k=1}^b w_k / T \right), \quad (8)$$

kde  $T$  je nastavený práh (v [10] doporučeno 0,7),  $w_k$  je váha k-tého rozložení v řadě (seřazené podle  $w/\sigma$ ) a  $B$  značí správné pozadí. Obecně je použito první rozložení díky

kterému  $\sum_{k=1}^b w_k$  bude větší než  $T$ . Následující obrázek ukazuje Gaussovska rozložení při adaptaci na blikající zadní světlo auta., kdy je vidět dvě rozložení jedno pro rozsvícené a druhé pro zhasínající. Díky zachování obou rozložení se model může mezi nimi rychleji přepínat.



Obr. 4 Rozložení Gaussianů při adaptaci na dvě různá pozadí

Jediným problémem zůstává jak nastavit počáteční střední hodnotu a odchylku na úplném začátku, v prvním snímku. Ne vždy je výhodné použít hodnoty z prvního snímku nebo prázdné pozadí. [9] [10] [11]

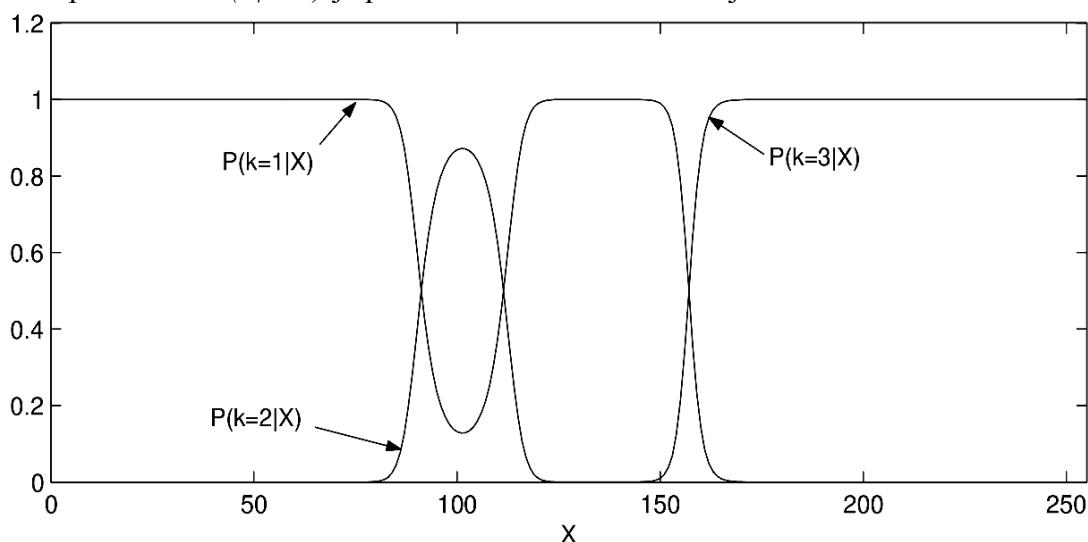
Výhody: dokáže zpracovat pohyb v pozadí a velmi rychle na něj reagovat změnou aktuálního pozadí a přitom se neztrácí informace o předchozím pozadí. Tak může být rychle přepínáno mezi jednotlivými pozadími, aniž by se muselo čekat než se model znovu adaptuje; vysoká kvalita výsledku, díky více Gaussovým rozložením.

Nevýhody: složitost algoritmu (byly nalezeny i mnohem složitější); výpočetní náročnost; pomalá prvotní stabilizace parametrů

Vylepšení: možnost použití barev. K nastavení střední hodnoty a odchylky na úplném začátku je doporučeno použít medián z několika snímků. Počáteční pomalá stabilizace se zrychlí použitím časově proměnného  $\alpha$ . Lze také použít prahování z hysterezí, kdy jsou použity dva prahy do (4) místo použité hodnoty 2,5. viz. [10]

Pro hlubší pochopení doporučuji [10]. Např. uvádí, že původní Background Mixture Model pracuje s pravděpodobnostním přístupem. Místo zmiňovaného „match“ (označujícího příslušnost aktuální hodnoty pixelu k danému Gaussianu) se vypočítává pravděpodobnost  $P(k|X, \Phi)$ , že pixel přísluší k danému Gaussianu; kde „ $k$ “ označuje pořadí Gaussianu,  $X$  aktuální pixel a  $\Phi$  vektor parametrů Gaussianu  $\mu$ ,  $\sigma$  a  $w$ .

Na základě této pravděpodobnosti se upraví parametry Gaussianů. Z výpočetních důvodů byla pravděpodobnost aproximována na 1 nebo 0 „match“, „non-match“ a také provedeno několik další zjednodušení, které zrychlují výpočet. Použitím aproximace příliš nezhoršuje přesnost algoritmu, protože jak je vidět z následujícího obrázku pravděpodobnost  $P(k|X, \Phi)$  je pro většinu hodnot  $X$  blízká jedné nebo blízká nule.



Obr. 5 Vysvětlení k aproximaci pravděpodobnosti  $P(k|X, \Phi)$  [10]

kde na ose  $y$  je pravděpodobnost  $P(k|X, \Phi)$ .

Článek dále na rozdíl od původního popisu MoG uvádí vhodné nastavení parametrů pro 3 Gaussiany a doporučuje několik vylepšení, které jsou samotnému algoritmu skoro nezbytné.

### 3.5 Kernel denstiy estimation

Jedná se o dále zobecněný model MoG. Je založen na větším bufferu vzorků a odhadu pravděpodobnosti pomocí příslušnosti pixelu k popředí pomocí všech těchto hodnot. Nejprve jsou uloženy vzorky  $x_1, x_2, \dots, x_N$ . Pravděpodobnost pro následující vzorek se vypočte jako:

$$\Pr(x_t) = \frac{1}{N} \sum_{i=1}^N K(x_i - x_t) \quad (9)$$

kde  $x_i$  je  $i$ -tý vzorek z bufferu,  $x_t$  je pixel aktuálního obrázku Iakt a  $K$  je zvolená funkce. Za  $K$  je dosazeno normální rozdělení, dostaneme:

$$\Pr(x_t) = \frac{1}{N} \sum_{i=1}^N \frac{1}{(2\pi)^{d/2} |\Sigma|^{d/2}} e^{-\frac{1}{2}(x_i - x_t)^T \Sigma^{-1} (x_i - x_t)} \quad (10)$$

kde  $\sigma$  je směrodatná odchylka a  $d$  počet barevných kanálů. Za předpokladu nezávislosti barevných kanálů se vzorec zjednoduší. Za jiného předpokladu použití šedotónového videa získáme:

$$\Pr(x_t) = \frac{1}{N} \sum_{i=1}^N \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{1}{2} \frac{(x_i - x_t)^2}{\sigma^2}} \quad (11)$$

Aktuální pixel  $x_t$  je označen jako popředí, pokud  $\Pr(x_t) < práh$ . Ze vzorce je vidět, že každá hodnota v bufferu je pak samostatným Gaussovským rozložením, na rozdíl od MoG kde je pro každý pixel použita jedno rozložení.

Uvedená lze velmi rychle vypočítat použitím přepočítaná look-up table, která je vytvořena pro všechny kombinace  $(x_i - x_t)^2$  a všechny možnosti  $\sigma$ . Jedná se nevelkou matici s rozměry  $256 \times Y$ , kde  $Y$  závisí na použitém rozlišení  $\sigma$ . Takováto konstrukce je ale obtížně, vypočítatelná v Matlabu, kde musí být použit maticový výpočet. Dalšího

zrychlení dosáhneme, pokud nejprve počítáme pouze část sumy  $\sum_{i=1}^N$ . Je velká

pravděpodobnost, že pro např.  $N=N/2$  bude pro většinu pixelů přesážen práh (v rovnici  $\Pr(x_t) < práh$ ). Pixel bude moc být zařazen do pozadí. Poté se pro zbývající pixely dopočítá celá suma. Směrodatná odchylka  $\sigma$  je odhadnuta pomocí.

$$\sigma = \frac{m}{0.68\sqrt{2}}, \quad (12)$$

kde  $m$  je medián ze všech možností  $|x_i - x_{i+1}|$ . To vychází z předpokladu, že vzorky přicházející v čase  $t=i$  a  $t=i+1$  se od sebe liší jen o úroveň šumu, ne díky přítomnosti jiného objektu nebo popředí. Mediánem pro všechny hodnoty z bufferu je odhadnuta úroveň šumu pro celý buffer.

### Řešení otřesů kamery

Metoda dále předpokládá parazitní pohyb kamery. K potlačení tohoto jevu není použita optická stabilizace, ale drobné otřesy jsou přímo zahrnuty do modelu. Předpokládá se, že pokud pixel nenáleží do pozadí na své aktuální pozici  $(x,y)$ , může se (vlivem posunu kamery) správné pozadí pro daný pixel posunout do jeho okolí, např. na pozici  $(x+1,y+1)$ . Je použito kruhového okolí o velikosti 5 pixelů.[18]. Celková pravděpodobnost libovolného posunu je dána jako

$$P_{\eta}(x_t) = \max_{z \in \eta(x)} \Pr(x_t | B_z) \quad (13)$$

kde  $\eta$  značí okolí pixelu  $x_t$ ,  $z$  pozici v daném okolí a  $B_z$  buffer pixelu  $v$  na pozici  $z$ . Pro každý pixel je tedy spočtena pravděpodobnost,  $Pr$ , ale jsou do ní dosazovány hodnoty bufferu na pozicích v okolí tohoto pixelu pro každou pozici v okolí. Pokud tato pravděpodobnost  $P_{\eta}$  je větší než nastavený práh nejedná se o popředí. Použitím pouze této pravděpodobnosti pro každý pixel je možné ztratit část popředí. O skutečný posun se jedná pouze v případě, když se posune celý spojitý objekt. Proto je dále vypočtena pravděpodobnost pro  $P_c$  každý spojitý objekt v popředí:

$$P_c = \prod_{x \in C} P_{\eta}(x) \quad (14)$$

$x$  značí všechny pixely ve spojitě oblasti  $C$ . Pro rozhodnutí je použit pravidlo  $(P_{\eta} > \text{práh}) \wedge (P_c > \text{práh2})$ , pokud je splněno pixel náleží do pozadí.

### Obnovování pozadí

Jako další vylepšení jsou použity dva modely pozadí – krátkodobý a dlouhodobý. Tím je řešen problém spojený s obnovováním bufferu. Je možné do něj přidávat pouze hodnoty příslušející k pozadí. Pokud se tam ale takto dostane pixel z popředí může způsobit chybnou detekci v následujících obrázcích, vzniká deadlock. Nebo je možné přidávat do bufferu každý pixel bez ohledu jeho příslušnost k popředí nebo pozadí. Tím ale dostáváme méně přesnou detekci pozadí.

Krátkodobý model se obnovuje jen pixely z pozadí. Do dlouhodobého je pak přidán každý nový pixel. Průnikem obou těchto modelů potlačíme obě výše popsání chyby a zlepšíme kvalitu pozadí.

Metoda dále řeší detekci stínů, pomocí barevného modelu rgb a přináší možnost přesného srovnání různých modelů pozadí. Detekce stínů je bude dále popsána v kapitole Barevné modely. Srovnání modelu je popisováno v kapitole Hodnocení algoritmů. V tomto hodnocení je mimo jiné uvedeno, že algoritmus vykazuje přesnější výsledky [18]

### 3.6 Jiné způsoby detekce pozadí

Při literárním průzkumu bylo nalezeno mnohem více možností detekce pozadí. Jednou z výpočetně nejnáročnější, ale také pravděpodobně nejkvalitnějších je výpočet optického toku.

Jeho zjednodušením je určovat pohyb ve scéně na pomoci významných bodů. Na každém snímku je vyhledáno větší množství významných bodů. Ty musí být nalezeny rovnoměrně po celém obrázku. Jsou sestaveny korespondence a na základě jejich posunu vytvořeno „řídce rychlostní pole“, podobně jako při výpočtu optického toku. [3]

Tímto způsobem vypočtené rychlostní pole není pro každý pixel. Potom lze použít některou ze segmentačních metod. Obraz segmentujeme ideálně na části odpovídající objektům ve scéně. Přesná segmentace není podmínkou, pokud na každém segmentu je nalezen alespoň jeden nebo lépe několik významných bodů. Pak je možné u každého segmentu určit jestli se pohybuje. Pokud, ale není na některém z pohybujících se objektů nalezen ani jeden významný bod metoda může selhat.

Mezi další možnosti detekce pozadí patří např. použití Local Binary patern. Ty jsou založeny na přepsání hodnot jednotlivých pixelů podle určitého pravidla, v závislosti na jejich okolí. Z tohoto nového „obrázku“ lze detekovat pohyb porovnáním histogramu Iakt a Iold (předchozího snímku). Pokud chceme zjistit, kde pohyb nastal lze obrázek rozdělit na velké množství oblastí. Výhodou této metody je její rychlost a kvalita při překvapivé jednoduchosti ; metoda je odolná na rychlé i pomalé změny osvětlení ve scéně a částečně i na malý pohyb pozadí. Nevýhodou je, že nově vypočtené hodnoty postrádají „fyzikální“ význam. [12]

Existuje velké množství další metod např. Eigenbackgrounds,...[8] Byl také prostudován algoritmus založený na cooccurrence of image variations, ale díky jeho nižší průhlednosti a obecně složitosti byl zamítnut.[35]

### 3.7 Různá obecná zlepšení

V předchozí kapitola popsane nebo jmenované metody jsou základem pro detekci pozadí. Dále je možné doplnit o různá vylepšení řešící obecné nebo konkrétní problémy.

#### Zrychlení

Obecně jakákoliv metoda lze zrychlit použitím snímků s nižším rozlišením. Při testech byl použit formát videa 640x480 a při stabilním fotoaparátu byly výsledky podobné testům se jednotlivými fotkami v mnohem větším rozlišení 2560x1920. Podle [12] je možné při označování míst kde nastal pohyb vynechávat sudé řádky. Vzniklé mezery se pak zaplní morfologickou operací uzavření s vhodným strukturním elementem nebo opakováním použitých řádků.

Pro detekci stínů lze využít barvu. Stín nemění barvu objektu na dopadá, změní se jen jeho jasová složka. Převedením do barevného modelu HSV se výpočet zjednoduší. Místa kde se nezmění barva, ale změní se jas lze detekovat jako stín. Pak se použije průnik s plochami označující pohyb ve scéně a dostaneme oblasti kde je stín pohybujícího se objektu. Tento postup selhává na pixelech, které jsou díky stínu tak tmavé, že jsou vyhodnoceny jako jiná barva.[14]

V [7] je popsán alternativní postup (k použití morfologického uzavření) k zaplnění „děravých“ oblastí vzniklých, když jsou jako pohybující se objekt určeny je okraje skutečného objektu. Malé díry jsou zaplněny takto. Vytvoří se okno 3x3 nebo 5x5. Střed okna se nastaví na každý pixel kde byl detekován pohyb. Pokud zbytek okna obsahuje víc než polovinu pixelů popředí, je i zbytek pixelů označen jako popředí. Pro velké díry se použije okno 25x25. Střed okna se umístí na každý pixel pozadí. V každém z 8 možných směrů se testuje jestli okno obsahuje dostatek pixelů popředí. Pokud ano pixel na středu je označen jako popředí.

### 3.8 Hodnocení algoritmů

Zatímco výpočetní a paměťová náročnost lze dobře odhadnout nebo změřit vyhodnocení přesnost je relativně problematické. Některé algoritmy mezi sebou ani není možné korektně porovnávat. Např. running gaussian average z principu neumí zpracovat periodicky opakovaný pohyb v pozadí, proto by nebylo korektní jej porovnávat s Gaussian Mixture modelem. Přesnost tohoto modelu zase závisí na nastavených parametrech, které mohou být pro některou scénu vhodnější než pro jinou. Navíc většina modelů používá kromě hlavního algoritmu detekce pozadí i další post-processing nebo různá vylepšení. Tyto dodatečné úpravy velmi silně ovlivňují výsledek a mohou být použity prakticky u všech algoritmů. Pokud by byly algoritmy testovány na reálné scéně, je nutné ručně vyhodnotit které pixely patří k pozadí. [8]

Tento problém je možné obejít vložení uměle definovaného objektu, známého tvaru do sekvence snímků tak, aby se tento „objekt“ ve videu pohyboval. Potom je snadné testovat pixely na kterých se v daném snímku má objekt vyskytovat, jestli jsou všechny označeny jako popředí. Pro porovnání více algoritmů lze použít toto: Scénu je nasnímána bez pohybu definovatelného jako popředí. Oba porovnávané modely se nastaví na shodnou chybovost (falešně detekované pixely jako popředí). Poté je vložen do videa uměle definovatelný objekt a je porovnávána přesnost detekování tohoto objektu různými metodami. Intenzitu objektu je dále možné měnit, od naprosto černého, přes šedou do bílé porovnávat schopnost detekce popředí pro různě velký kontrast objektu a pozadí. [18]. Na druhou stranu, jak bylo napsáno výše, některá srovnání nemají příliš velkou vypovídací hodnotu.

### 3.9 Použití barevného videa

Původní algoritmy detekce pozadí byly z důvodu obecnosti navrženy pro černobílou kameru. Byly očekávány lepší výsledky při detekci v některém z barevných prostorů. Další použití barvy může být k vytvoření jasově invariantního barevného prostoru k odstranění problémů s náhlou změnou osvětlení, nebo pro výše popsanou detekci stínu. Postupně bylo vyzkoušeno několik barevných prostorů, a zjištěna jejich vhodnost pro tento účel.

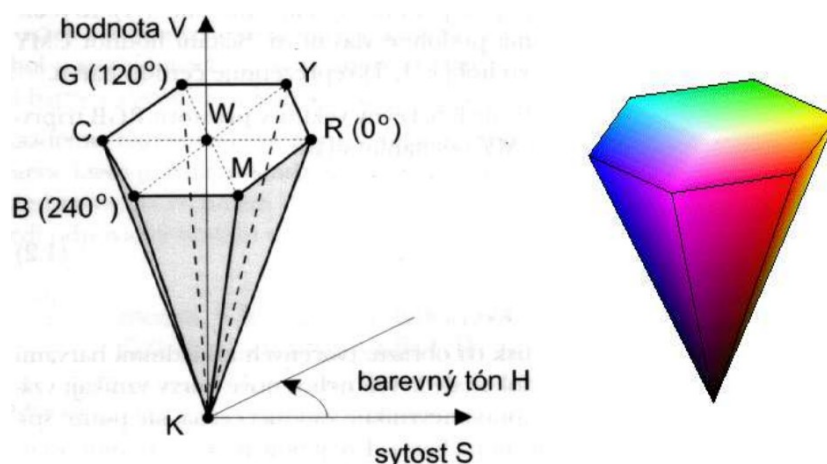
#### 3.9.1 Barevné modely

RGB je nejpoužívanější barevný model. Barvu rozkládá do 3 složek R-Red, G-Green a B-Blue. Je zobrazován jako barevná krychle s maximy R,G,B barev ve 3 jejích rozích. V tomto modelu pracují videokamery, takže je přirozené jej použít i pro zobrazování. Na druhou stranu se nejedná o prostor nejlépe simulující lidské vidění a je také méně vhodný pro představu a tvorbu barev (na rozdíl od jiných modelů). Barvy v tomto prostoru nejsou invariantní na změnu osvětlení nebo stín na objektu. Proto může být prostor normalizován a vytvořen tzv. rgb barevný prostor.

$$r = \frac{R}{R + G + B} \quad g = \frac{G}{R + G + B} \quad b = \frac{B}{R + G + B} \quad (15)$$

V tomto prostoru jsou složky rgb na zmíněné změny invariantní. Ani toto řešení není ideální. Problémy nastávají na tmavých plochách, kde  $R+G+B$  se blíží nule. Kromě možného dělení nulou je na těchto plochách velmi výrazný vliv šumu, který tak mění barvu objektů. Na tmavých plochách je proto zbytečné se zabývat invariantností nebo spíše barevnou informací obecně. Druhý popisovaný problém je nedostatečná sytost barev reálných objektů[1]. Oba problémy byly z vlastních experimentů potvrzeny. [19]

Dalším používaným modelem je HSV. Barvu rozkládá na Hue – barevný tón, Saturation – sytost barvy a Value – jasová hodnota. Následující obrázek znázorňuje tento model.



Obr. 6 Geometrická reprezentace barevného modelu HSV [13]

Z obrázku je vidět jeho přednosti a nevýhody. Barva (barevný tón) v tomto prostoru je invariantní na změnu osvětlení a stín. Model je také snadno představitelný. Lépe popisuje lidské vidění, při nízkém osvětlení (složka  $V$ ) oko obtížněji rozeznává jednotlivé barvy.

Menší nevýhodou může být pohyb hodnot barevného tónu  $H$  po šestiúhelníku při konstantní sytosti barvy  $S$ . [13] Větším nedostatkem, hlavně při metodách odečítání pozadí je „periodičnost“ barevného tónu  $H$ . Model má jeho hodnoty vyjádřené jako 0-1. Malé hodnoty vyjadřují nejprve červenou, pak oranžovou a ostatní barvy, potom zase zpátky k červené. Problém je možné vidět i naopak pro červenou barvu existují dvě (velmi rozdílné) hodnoty složky  $H$  (blízká nule a blízká jedničce). Navrhovaná řešení jsou popsána v části o implementaci algoritmů.

Dále bylo použito několik dalších barevných modelů pro porovnání, jejich vlastnosti zde nebudou popisovány.

### 3.9.2 Měření podobnosti barev

Většina algoritmů detekce pozadí rozhoduje o pohybu ve scéně na základě rozdílu hodnot jasu, tomto případě barvy. Existuje několik možností jak zjistit podobnost v barevném prostoru. [19]

#### Euklidovská vzdálenost

Jedná se o nejjednodušší metodu, popsatelnou např. pro RGB prostor vzorcem

$$D = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2} \quad (16)$$

kde  $R_1, G_1, B_1$  jsou hodnoty barevných složek prvního a  $R_2, G_2, B_2$  hodnoty druhého pixelu. Tento způsob v prostoru RGB není invariantní na změnu jasu.

#### Úhel vektoru

Teoreticky vhodnější pro tento barevný prostor je výpočet podobnosti na základě úhlu vektoru. Tento úhel se nemění při změně osvětlení jednoho z pixelů. Změna osvětlení se projeví posunem všech barevných složek v RGB krychli, ale směr zůstane stejný. Úhel dvou vektorů lze vypočítat podle:

$$\cos \theta = \frac{\vec{v}_1^T \vec{v}_2}{\|\vec{v}_1\| \|\vec{v}_2\|}, \quad (17)$$

kde  $\vec{v}_x$  je vektor udaný hodnotami RGB složek a  $\|\vec{v}_x\|$  označuje jeho normu. Vhodnější způsob výpočtu, který má vyšší rozlišitelnost při malých úhlech popisuje následující vzorec:

$$\sin \theta = \sqrt{1 - \left( \frac{\vec{v}_1^T \vec{v}_2}{\|\vec{v}_1\| \|\vec{v}_2\|} \right)^2} \quad (18)$$



Metoda vykazuje dobré výsledky ve většině barevných prostorech. [19] [20]

## 4 POHYBUJÍCÍ SE KAMERA

Pohyb kamery může nastat záměrně (např. kamera připevněná na sloupu uprostřed parkoviště se bude otáčet kolem vlastní osy) nebo nechtěně (na dlouhém nosníku, kterým kýve vítr). Kamera se může pohybovat i pokud je připevněná na velkém statickém objektu. Důkazem je video dálnice na mostě, přes který chodili lidé. Ve videu jsou vidět zřetelné otřesy. I pokud ve videu není chvění vidět, projevuje se v rozdílovém snímku jako detekce pohybu na hranách, kde dochází k velkému rozdílu jasových nebo barevných hodnot. To nemusí být na výsledném pozadí patrné, ale zhoršuje se kvalita popředí..

Nechtěný pohyb lze pro statickou kameru vyřešit mechanicky. V případě otáčející se kamery na sloupu, nebo kamery umístěné v autě, které jednou za čas projíždí zvolenou trasu je potřeba dále správně spojit jednotlivá natáčení videa

### 4.1 Sesouhlasení snímků

Sesouhlasení (lícování, registrace) více obrázků znamená nalezení správného posunu (obecně libovolné geometrické transformace) následujícího obrázku tak, aby po provedení tohoto posuvu se zvolené části obrázku překrývaly s výchozím obrázkem. To může být použito např. pro sledování pohybu kamery nebo objektu, napojování částí panoramatických snímků, rekonstrukci 3D tvaru ze snímků z více kamer,... [21] V této práci je využito hlavně k optické stabilizaci obrazu. Pro sesouhlasení je možné použít různé metody, nejčastější je lícování pomocí významných bodů, vyhledání vzorů (template matching), fázovou korelaci nebo pomocí sdruženého histogramu. [24]

### 4.2 Metoda využívající korelace

Postup využívá korelace mezi částí výchozího obrázku a všemi dalšími následujícími obrázky. Ve výchozím obrázku je vybrána šablona. Na ni je kladeno několik požadavků. Měla by být dobře rozpoznatelná, vyskytovat se na obrázku nejlépe pouze jednou (případně by vícenásobné výskyty měli být od sebe dostatečně vzdáleny) a nacházet se ve středu obrázku (na okrajích hrozí že z obrázku při velkém posuvu kamery zmizí). Díky těmto požadavkům je obtížné automatizovat výběr takového objektu. Šablona je proto nejprve zvolena uživatelem. Tato metoda byla vytvořena pro statickou nebo otočnou kameru na parkovištích nebo na ulici. Uživatel by pouze při spuštění vybral referenční objekt a program by už nadále pracoval automaticky. Jako vhodný objekt lze použít dopravní značku, nebo dobře viditelné číslo popisné na domě.

Na šablonu (masku) je použit hranový detektor. Každý následující obrázek je také filtrován hranovým detektorem a je provedena korelace. Nejvyšší hodnota v „korelačním obrázku“ udává pozici šablony. Z rozdílu polohy šablony ve výchozím obrázku a v každém následujícím se vypočítá posun. Každý obrázek se pak posune opačným směrem.

Následující vzorec uvádí formální výpočet.

$$CC(x', y') = \sum_{\substack{x, y \in T \\ x+x', y+y' \in F}} (f(x+x', y+y') * t(x, y)) \quad (19)$$

kde funkce  $f(F)$  je obrázek, funkce  $t(T)$  je šablona. Korelace může vykazovat vysoké hodnoty v „korelačním obrázku“ bez ohledu na správné umístění objektu. To nastane když jsou hodnoty obrázku i šablony větší než nula, případně výrazně větší než nula (bílé oblasti v obrázcích).[22] Tento problém byl částečně vyřešen použitím výše zmíněného hranového filtru.

Metodu lze výrazně zrychlit (experimentálně ověřeno) při použití korelace ve frekvenční oblasti. Pro malé obrázky a šablony je rychlejší standardní korelace. Od určité velikosti je rychlejší použití korelaci ve frekvenční oblasti. Masku se otočí o 180°. Otočením masky se korelace převede na konvoluci. [23] Otočená maska i obrázek se převede Fourierovou transformací na spektrum. Výpočetně náročné konvoluci v obrazové oblasti odpovídá ve spektrální oblasti jednoduché násobení. Výsledek násobení je pak převeden zpětnou Fourierovou transformací zpět na obrázek. Nejvyšší hodnota ve výsledném obrázku opět udává polohu šablony. [23]

Standardní korelace ani korelace ve spektrální oblasti není invariantní na různé změny jasu ani otočení masky. Podle mé úvahy, potvrzené v [24], využití hranového detektoru před výpočtem korelace snižuje vliv změn jasu. Rotace masky se případech v této práci nevyskytuje.

V testech bylo zjištěno, že metoda vyhledá konkrétní číslo popisné, i když je jich na snímku víc stejného i různého typu. Také se ale stalo, že maximální hodnota „korelačního obrázku“ byla dočasně mimo správné umístění. Toto lze vyřešit omezením maximálního posunu od výchozího obrázku. Pokud bude překročen použije se dočasně poslední správný posun.

Metodu by bylo možné vylepšit použitím barevné místo šedotónového obrázku. Např. při výskytu výše popsané chyby testovat průměrnou barvu oblasti nalezené v následujícím obrázku, jestli je alespoň podobná průměrné barvě objektu. Dalším zpřesněním by bylo vyhledávání 2 objektů (které by se dostatečně lišily navzájem a od zbytku obrázku). Z nich by se získaly dvě možné korekce posunu. Pokud by u jednoho z objektů byl překročen maximální posuv, ke správnému posunu obrázku by se použilo druhého objektu. To by zabránilo dočasnému výpadku stabilizace obrazu.

### 4.2.1 Normalizovaná korelace

Jak je popsáno výše, standardní korelace dává dobré výsledky na čistě černobílých obrázcích nebo na obrázcích s velmi výraznými hranami a co nejhomogennějšími plochami mezi nimi. Zvýraznění hran funguje jako vysokofrekvenční filtr, který odstraní střední hodnotu. Pokud zvolíme nízkou mezní frekvenci může střední hodnota částečně zůstat. Šablona potom není nalezena na správném místě. Pokud zvolíme příliš vysokou mezní frekvenci, můžeme odstranit i některé hrany obrazu.[33] Normalizovaná korelace řeší tento problém. Od masky i obrazu se odečte jejich střední hodnota. Dále se v každém bodě se korelace normalizuje rozptylem v celém obrázku a lokálním rozptylem pod maskou. Definovaná je vzorcem:

$$NCC(x', y') = \frac{1}{N_1} \sum_{\substack{x, y \in T \\ x+x', y+y' \in F}} ((f(x+x', y+y') - \bar{f}(x, y)) * (t(x, y) - \bar{t}(x, y))) \quad (20)$$

$$N_1 = \sqrt{D_f(x', y') * D_t}$$

$$D_f(x', y') = \sum_{x+x', y+y' \in F} ((f(x+x', y+y') - \bar{f}(x, y))^2$$

$$D_t = \sum_{x, y \in T} (t(x, y) - \bar{t})^2,$$

kde  $\bar{f}$  znamená střední hodnotu obrazu,  $\bar{t}$  střední hodnotu šablony,  $D_f$  lokální rozptyl v obraze a  $D_t$  lokální rozptyl masky.[22]

Problémem přímého použití tohoto vzorce je velká časová náročnost. Následujícími úpravami lze výpočet urychlit. V čitateli lze od obrazu i od šablony odečíst střední hodnotu.

$$NCC(x', y') = \frac{1}{N_1} \sum_{\substack{x, y \in T \\ x+x', y+y' \in F}} (f'(x+x', y+y') * t'(x, y)) \quad (21)$$

Potom lze s výrazem v čitateli provést korelaci ve frekvenční oblasti. Člen  $D_t$  v čitateli lze předpočítat dopředu, nebo šablonu normalizovat aby její rozptyl byl roven jedné.

Člen  $D_f$  lze vyjádřit jako lokální sumu:

$$e_f(u, v) = s^2(u+N-1, v+N-1) - s^2(u-1, v+N-1) - s^2(u+N-1, v-1) + s^2(u-1, v-1) \quad (22)$$

$$s(u, v) = f(u, v) + s(u-1, v) + s(u, v-1) - s(u-1, v-1)$$

$$s^2(u, v) = f^2(u, v) + s^2(u-1, v) + s^2(u, v-1) - s^2(u-1, v-1)$$

kde  $N$  je velikost obrázku. Funkce  $s(u, v)$  i  $s^2(u, v)$  je nulová pro  $u, v < 0$ . Normalizovaná korelace je pak navíc invariantní na změnu celkového osvětlení ve scéně a její hodnoty, na rozdíl od obyčejné korelace, jsou v rozsahu 0-1. [33]

## 4.3 Metoda využívající významných bodů

Sesouhlasení je provedeno na základě bodů, které ve skutečných obrazech označují shodné místo. V každém obrázku jsou nalezeny tzv. významné body, které se často shodují s rohy nebo hranami. Dále jsou vyhledány odpovídající si páry těchto bodů. Ze různých souřadnic významných bodů v obou obrazech je vypočítán posun

### 4.3.1 Detektory významných bodů

Významné body (lokální deskriptory, rohové body) jsou takové body, které se co nejméně podobají svému nejbližšímu okolí a které tak lze v jejich blízkém okolí snadno lokalizovat. Formálně je možné je definovat jako body mající ve svém definovaném okolí vysoký gradient jasové funkce. Takovými body jsou hlavně rohy. Rohy lze totiž v jejich okolí lokalizovat v obou směrech. Dalšími významnými body mohou na základě definice být konce přímek v obraze, špičky nebo osamocené body odlišného jasu. K jejich nalezení slouží detektory rohů. Měli by mít tyto vlastnosti:

- detekovat co nejvíc skutečných rohů
- co nejméně falešných rohů
- odolnost na šum
- invariantnost na změny změny jasu
- invariantnost na geometrické transformace obrazu (posun, změna měřítka,...)
- nízká výpočetní náročnost

Žádný detektor rohů nefunguje ideálně. Postupně bylo vynalezeno mnoho různých detektorů s různými zlepšeními. Podrobný výpis a nalezení toho nejlepšího by vydalo na samostatnou práci, jak je vidět z [25]. Jeden z používaných operátorů je Harrisův operátor [26]. [21]

### 4.3.2 Moravcův operátor

Jedná o první nejjednodušší rohový detektor. Je definován tímto vzorcem:

$$f(x, y) = \frac{1}{8} \sum_{k=x-1}^{x+1} \sum_{e=y-1}^{y+1} |g(k, e) - g(x, y)|, \quad (23)$$

kde  $x, y$  označují střed masky  $3 \times 3$ . Operátor prochází maskou celý obrázek a počítá v podstatě průměr rozdílů středového pixelu a každého pixelu z 8-okolí. Pokud je operátor nad hladkou plochou bez jasových změn je výsledek nulový. Postupně větší odezvu dává operátor pro hrany, pak rohy a nakonec osamocené odlišné pixely. [21] Výsledek toho operátoru je dost nepřesný.

### 4.3.3 Harrisův detektor

Tento detektor je dnes velmi často používaný. Obrázek se prochází maskou. Nad každým pixelem je maska posunuta do všech směrů. Na hladké ploše nejsou v žádném

směru zaznamenány změny intenzity. Na hraně je změna zaznamenána v směru kolmém na hranu. Na rohu je změna zaznamenána ve všech směrech. To formálně odpovídá autokorelační funkci. Ta je definována jako

$$c(x, y) = \sum_{u, v \in W} [I(u, v) - I(u + \Delta x, v + \Delta y)]^2, \quad (24)$$

kde  $x, y$  jsou souřadnice v obrázku  $u, v$  jsou souřadnice v masce,  $I$  označuje jasovou funkci (obrázek),  $W$  masku a  $\Delta x, \Delta y$  posun masky. Funkce s posuvem masky se aproximuje Taylorovým rozvojem prvního řádu.

$$I(u + \Delta x, v + \Delta y) \approx I(u, v) + [I_x(u, v) \quad I_y(u, v)] * \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \quad (25)$$

kde  $I_x(u, v)$  je parciální derivace obrázku ve směru  $x$  a  $I_y(u, v)$  je parciální derivace ve směru  $y$ . Tyto derivace se pro jejich výpočet aproximují diferencí. Dosazením do vzorce (24) dostaneme:

$$c(x, y) = \sum_{u, v} [I(u, v) - I(u, v) - [I_x(u, v) \quad I_y(u, v)] * \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}]^2 \quad (26)$$

Po odečtení  $I(u, v)$  a roznásobení matic a umocnění na druhou:

$$c(x, y) = \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} * \begin{bmatrix} \sum_w I_x^2(u, v) & \sum_w I_x(u, v) * I_y(u, v) \\ \sum_w I_x(u, v) * I_y(u, v) & \sum_w I_y^2(u, v) \end{bmatrix} * \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (27)$$

Pro zjednodušení zápisu se provede označení

$$c(x, y) = \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} * C(x, y) * \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (28)$$

V tomto vzorci je důležitá matice  $C(x, y)$ . Vlastní čísla této matice  $\lambda_1, \lambda_2$  se vypočítají z rovnice:

$$\det \left( \begin{bmatrix} I_x^2 - \lambda & I_x * I_y \\ I_x * I_y & I_y^2 - \lambda \end{bmatrix} \right) = 0, \quad (29)$$

Tyto vlastní čísla už samy o sobě vypovídají o povaze obrázku pod maskou.

Pro  $\lambda_1$  a  $\lambda_2$  malé se jedná o hladkou konstantní plochu

Pro jedno  $\lambda$  velké a druhé malé se jedná o hranu

Pro  $\lambda_1$  i  $\lambda_2$  velké se jedná o roh.

V Harrisově operátoru, je dále použit tzv. indikátor přítomnosti rohu, který dále pracuje s vlastními čísly, aby se vyhnul jejich časově delšímu výpočtu (řešení kvadratické rovnice) [27]

$$H(\lambda_1, \lambda_2) = \det(C(x, y)) - \kappa * \text{trace}(C(x, y)) \quad (30)$$

$$H(\lambda_1, \lambda_2) = \lambda_1 * \lambda_2 + \kappa * (\lambda_1 + \lambda_2) \quad (31)$$

$$H(\lambda_1, \lambda_2) = (C_{11} * C_{22} - C_{12} * C_{21}) + \kappa * (C_{11} + C_{22}) \quad (32)$$

kde  $\kappa$  je empirická konstanta v intervalu 0,04 až 0,15. Vyšší hodnota konstanty  $\kappa$  lépe rozpoznává rohy od hran, ale označuje jako rohy i více konstantní oblasti (relativně oproti nižší hodnotě konstanty).

Protože je využita derivace, je operátor náchylný na šum. Proto se musí před jeho použitím obrázky filtrovat např. Gaussovým filtrem. Pro zrychlení lze využít separabilních vlastností Gaussova filtru. Po získání odezvy Harrisona operátoru  $H(\lambda_1, \lambda_2)$ , se vyhledají lokální maxima, např. jednoduchým porovnáním hodnoty pixelu s hodnotami v jeho 8-okolí. Výsledná lokální maxima se prahují zvoleným prahem. Ten určí výsledný počet nalezených rohových bodů. Ty se pak už jen vykreslí do obrázku.

Harrisův operátor má několik výhod kvůli kterým se používá. Je to relativně dobrá přesnost, nezávislost na posuvu, na rotaci obrázku (vyplývá z použití vlastních čísel matice), na změnách intenzity, malých změnách snímání a dobrá výpočetní rychlost. Současně má i několik nevýhod závislost na změně měřítka a na velkých změnách kontrastu v obrázku, dále výsledek závisí na větším množství parametrů (parametry filtru proti šumu a volba prahu). Takže získání dobrého výsledku nelze automatizovat. Pro automatizované použití je třeba vždy zvolit nějaký kompromis. [21]

Harrisův detektor je relativně rychlý, ale oproti jiným algoritmům méně přesný. Proto bylo postupně vytvořeno několik modifikací. Např. Shi-Tomasi algoritmus navrhuje vlastní čísla vypočítat (rovnici (33)) a použít jako odezvu detektoru  $H(\lambda_1, \lambda_2)$  menší z obou vlastních čísel.

$$\lambda_{1,2} = \frac{I_x^2 + I_y^2 \pm \sqrt{(I_x^2 - I_y^2)^2 + 4 * I_x I_y * I_x I_y}}{2} \quad (33)$$

Jiná úprava (Förstner) používá místo součtu ve výsledné rovnici odezvy (rovnice (32)) podíl. [26]

$$H(\lambda_1, \lambda_2) = \frac{\lambda_1 * \lambda_2}{\lambda_1 + \lambda_2} \Rightarrow H(\lambda_1, \lambda_2) = \frac{C_{11} * C_{22} - C_{12} * C_{21}}{C_{11} + C_{22}} \quad (34)$$

Další úprava (Haralickův operátor) vypočítává tzv. váhu (rohovosti)  $w$  jako

$$w = \det(C) = \lambda_1 * \lambda_2 = C_{11} * C_{22} - C_{12} * C_{21}, \quad (35)$$

a dále kruhovost  $q$ , která slouží k zamítnutí významných bodů na hranách. Pokud je kruhovost  $< 1$  jedná se o bod na hraně a je zamítnut, např. se nastaví váha na 0. [28]

$$q = 1 - \left( \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \right)^2 \quad (36)$$

Existují ještě další úpravy, ale vzhledem k existenci takového počtu úprav nemusí ani vyzkoušení další přinést lepší výsledky.

### **Možná obecná vylepšení**

Libovolný rohový detektor lze vylepšit několika způsoby. Není vhodné vyhledávat rohy v celém obrázku, ale jen v prostřední části. Rohy na krajích by mohli při posunu zmizet. To lze vyřešit na konci detekce, prostým odstraněním bodů blízko okraje. Druhou možností je „vyfiltrování okrajů“ silnějším Gaussovým filtrem. Pak teoreticky nebudou rohy na okrajích ani nalezeny nebo jich bude nalezeno méně.

Na skutečných obrázcích se také často většina bodů nahromadí na jednom objektu (navíc na tom, který nepatří do statického pozadí). Proto je lepší obrázek rozdělit na několik částí a v nich vyhledat rohové body. Takto vyhledané body budou potom rovnoměrněji rozmístěny po obraze a pokud není celý obraz zaplněn pohybujícími se objekty bude alespoň několik z nich umístěno ve statickém pozadí.[29]

Pokud nastavíme práh (pro odstranění bodů s nízkou hodnotou „rohovosti“) nevhodně může být na některém obrázku označeno velké množství rohů (z nichž pravděpodobně jich bude hodně falešných). Proto je vhodné omezit maximální počet nalezených bodů konstantou.

Další úprava také může vylepšit volbu prahu. Uložíme nejvyšší hodnotu „rohovosti“ a práh určíme jako „x“ procent z této hodnoty. [29]

### **4.3.4 Nalezení odpovídajících si významných bodů**

Druhým základním problémem sesouhlasení pomocí významných bodů je určit které body z výchozího obrázku patří ke kterým z následujícího obrázku.

Nejjednodušší možnost je přiřadit ke každému bodu z prvního obrázku ten z druhého obrázku, který má k němu nejmenší vzdálenost. Pokud je tato vzdálenost menší než nastavený práh jsou body určeny jako odpovídající si. Použitím prahu také s větší pravděpodobností vybereme body ze statického pozadí, protože jejich případný posun je způsoben jen otřesy kamery. Zatímco vzdálenost bodů nalezených na pohybujícím se objektu je větší o pohyb tohoto objektu a body jsou prahováním vyřazeny.

Přiřazením nejbližšího bodu, ale může nastat, že jeden bod z druhého obrázku patří k více bodům z prvního. Tento problém lze vyřešit tak, že po každém přiřazení, právě přiřazený bod odstraníme (např. nastavením jeho vzdálenosti na maximální hodnotu).

Další možností je najít k bodu z prvního obrázku nejbližší bod a druhý nejbližší. Pokud je druhý nejbližší bod dostatečně vzdálen od prvního nehrozí, že by nastala chyba. Pokud je druhý nejbližší bod blíže než např. 0,5 vzdálenosti bodu z prvního obrázku k nejbližšímu z druhému, jsou oba body zamítnuty, protože není jisté které přiřazení je to správné.[30]

Velmi dobrou metodou může přiřazení významných bodů na základě korelace jejich okolí s okolím bodů ve druhém obrázku. Tento vlastní nápad se později ukázal jako dobrý, protože podobná metoda je používána v [37], kde dále doporučují zjišťovat korelaci okolí jen mezi body do určité maximální vzdálenosti.

Po nalezení korespondencí všech významných bodů je z rozdílů souřadnic v obrázku vypočten pro každý pár bodů posuv. Za výsledný posuv je použit ten nejčastější.

### 4.3.5 RANSAC algoritmus (Random Sample Consensus)

Tento algoritmus řeší sesouhlasení bodů jiným způsobem. Nejsou vyhledávány odpovídající si body. Náhodně se opakovaně vytváří předpokládaný model geometrické transformace a ohodnocuje se jeho přesnost. Pak se vybere ten nejlepší. Pro tento konkrétní případ se postupuje v těchto krocích.

1. Vyber náhodně jeden bod z prvního obrázku a náhodně ho přiřaď k bodu z druhého obrázku.
2. Na základě těchto dvou bodů urči posuv mezi obrázky
3. Transformuj všechny body z druhého obrázku podle tohoto posunu
4. Urči kolik bodů z prvního obrázku odpovídá (s tolerancí odpovídá) tomuto posuvu. Počet odpovídajících bodů  $K$  ulož. K němu ulož hodnoty posunu.
5. Opakuj body 1 až 4.
6. Pokud  $K_{n+1} > K_n$  ulož nový posun a nové  $K$
7. Opakování ukonči, pokud bylo dosaženo stanovené přesnosti (počtu bodů  $K$ ) nebo proběhl nastavený počet opakování.

Pro velký počet bodů je metoda velmi robustní. Velkou nevýhodou je, neznámá maximální doba trvání algoritmu. [31] Další nevýhodou je nutnost volby požadované přesnosti (počtu bodů  $K$ ). To lze obejít nastavením  $K$  jako „ $x$ “ procent z celkového počtu bodů. Pokud ale bude většina bodů na pohybujícím se objektu, může se stát, že se obrázek bude posouvat podle pohybu tohoto objektu, ne podle otřesů kamery. Možný řešením může být výše zmiňované rozdělení obrázku na několik částí a v každé hledat významné body zvlášť. Tím bude zaručeno, že alespoň některé body padnou do statického pozadí. Je možné vyřadit z algoritmu ty body, které se nacházejí v oblasti, kde v předchozích snímcích nastal pohyb. Takto lze sesouhlasovat i obrázky se složitějšími geometrickými transformacemi než je posun. Stačí upravit vstupní „model“

## 4.4 Detekce pozadí při použití otáčející se kamery

Kromě statické kamery, stabilně snímající jedno místo se může kamera i pohybovat. Jeden z jednoduchých pohybů může být otáčení. Dalším může být kývání ve vodorovné rovině nebo libovolný jiný periodicky se opakující pohyb. Detekce pohybu z videa se pak stává složitější, protože musí být kompenzován tento pohyb.

Pro detekci pohybu se potřeba najít snímky zobrazující v různých periodách stejnou část scény. V nejjednodušším případě je k dispozici signál ze snímáče úhlové polohy. Pak stačí propojit aktuální snímek s příslušným snímkem podle externího údaje o natočení kamery. Když není přesnost snímáče dostatečná, je vhodné provést



sesouhlasení. Pokud se kamera otáčí konstantní rychlostí stačí detekovat průchod počáteční polohou a sesouhlasit vždy snímky s odpovídajícím pořadím v otočce.

Když nelze splnit ani jednu možnost může být detekce pozadí vyřešena následujícím způsobem, čistě softwarovým řešením. V průběhu první otočky kamery se postupně vybírají objekty. Nový objekt je vybrán, když se starý objekt dostane příliš blízko k okraji zorného pole kamery. Pak by hrozilo, že při větší rychlosti otáčení už na dalším snímku objekt nebude a současně nebude vybrán žádný jiný. Každý zvolený objekt se v průběhu jeho výskytu ve videu trasuje a zaznamenávají se jeho polohy. Trasování je vyřešeno pomocí template matchingu. Dále se zaznamenává o jaký šlo objekt a v jakém intervalu snímků se trasoval. V průběhu druhé a každé další otočky kamery se nejprve zjistí, který objekt se má ve snímku vyhledávat. Objekt je na snímku vyhledán a podle jeho polohy se najde nejpodobnější snímek z první otočky. Protože mohlo nastat určité posunutí může být následně provedeno sesouhlasení obou k sobě náležejících snímků.

Protože se jedná o rotační pohyb, může vnikat i určité geometrické zkreslení. To je možné odstranit pomocí lícovacích bodů a jejich korekcí. Na obou snímcích jsou nalezeny významné body. Následně se určí vzájemné korespondence bodů. Vypočte se transformační matice pro korekci zkreslení a ta se aplikuje. Problém vniká pokud je zkreslení v některé části obrázku příliš velké a není použit afině invariantní detektor významných bodů. Pak nemusí být významné body nalezeny přesně na stejných místech a zkreslení není kompenzováno dokonale, případně může být i zvětšeno.

## 4.5 Detekce pozadí z libovolně se pohybující kamery

Další možností, kromě statické kamery a otáčející se kamery je kamera umístěná v jedoucím autě. Auto projíždí opakovaně zvolenou trasu a automaticky je detekováno pozadí a pohybující se objekty. Klíčovou částí algoritmu je nalezení odpovídajících si snímků nebo odpovídajících si částí snímků na k sobě patřících snímcích. Celá plocha snímku si pravděpodobně nebude odpovídat nikdy protože auto nejedí po úplně stejných trajektoriích. Pokud jsou tyto odpovídající si snímky (části) nalezeny lze na ně použít jakýkoliv algoritmus na detekci pozadí.

První, jednodušší možností je hledat ke každému aktuálnímu snímku odpovídající si snímek z první nebo z předchozí jízdy. To lze provést na základě korespondence významných bodů. V každé jízdě jsou vyhledávány významné body afině invariantním detektorem např. afině invariantní Harrisův detektor. Ten je rozšířením klasického Harrisova detektoru na invariantnost vůči změně měřítka a vůči afiní transformaci obrazu. Významné body v obraze jsou pak nalezeny na stejné části objektu bez ohledu na úhel pohledu na objekt. Po vyhledání významných bodů je nutné nalézt jejich korespondence. Tento detektor vyhledává kromě významných bodů a také okolí těchto bodů. Toto okolí dále normalizuje na „kolmý pohled“ na objekt. Pro libovolný úhel pohledu na objekt je normalizované okolí vzhledově stejné. Okolí lze použít přímo pro

vyhledání shodných snímků, např. template matchingem nebo lze vypočítat deskriptory tohoto okolí a ty použít pro zjištění korespondence.

Druhou možností je porovnávat snímky v rámci jedné jízdy. Díky pohybu auta a tím kamery vzniká mezi dvěma následujícími snímky geometrické zkreslení. Opět jsou vyhledávány významné body afině invariantním detektorem. Je zjištěna jejich korespondence a následně jsou použity jako lícovací body pro geometrickou transformaci aktuálního obrázku tak, aby odpovídal obrázku předchozímu. Tímto způsobem lze získat pozadí a detekovat pohybující se objekty v rámci jedné jízdy. Afině invariantní Harrisův detektor je popsán např. v [34]

## 5 POPIS IMPLEMENTACE NĚKTERÝCH ALGORITMŮ A VÝSLEDKY TESTŮ

### 5.1 Detekce pozadí

V této kapitole budou popsány některé vybrané části vyzkoušených programů, obecně použité postupy a různé postřehy při testování.

#### 5.1.1 Obecně ke každé metodě

V každé použité metodě je pozadí aktualizováno, pokud v rozdílové snímku nenastal pohyb (daný pixel má hodnotu 0). Aktualizaci pozadí lze provést cyklem přes všechny pixely:

```
rozmetry=size(Ir_open);  
Ip=zeros(rozmetry);  
for i=1:rozmetry(1)*rozmetry(2)  
    %pokud nenastal pohyb  
    if Ir_open(i) == 0  
        Ip(i) = Iakt(i);  
    end;  
end;
```

který je ale výpočetně náročný a způsobuje viditelné zpomalení programu. V [8] uvádějí v rovnici (3) možné řešení:

$$Ip_{akt} = Ip_{akt} * (Ir) + (1 - Ir) * ((1 - \alpha) * Ip_{akt} + \alpha * Iakt), \quad (37)$$

kde  $Ip_{akt}$  je aktuální pozadí,  $Ir$  rozdílový prahovaný snímek,  $\alpha$  adaptační koeficient a  $Iakt$  aktuální snímek. Pokud nastal pohyb (příslušný pixel  $Ir=1$ ) nové pozadí se aktualizuje podle starého, protože druhý sčítanec se vynuluje. Pokud pohyb nenastal vynuluje se první sčítanec a pozadí se aktualizuje podle  $Iakt$ , nebo rovnou podle klouzavého průměru.

Podobně lze vylepšit prahování, místo použití cyklu

```
for i=1:size(a,2),  
    if a(i)<th1  
        c(i)=0;  
    elseif a(i)>th2  
        c(i)=255;  
    end  
end
```

Podle [17] je rychlejší tato konstrukce:

```
Ir(Ir<prah) = 0;  
Ir(Ir>prah) = 255;
```

Bylo zjištěno, že jako index nelze použít některé složitější podmínky.

Na začátku práce se vyskytl problém s načtením i docela krátkého videa do matlabu, i přesto, že videa zjevně nepřevyšovalo paměťové omezení udávané matlabem k dané chybě. To bylo vyřešeno načtením vždy jen jednoho snímku, místo celého videa

```
akt_snimek=aviread(nazev_video,i);
```

a počet snímků byl zjištěn příkazy:

```
video_info = aviinfo(nazev_video,'Robust');  
pocet_snimku=video_info.MainHeader.TotalFrames;
```

U libovolné metody je nutné prahovaný rozdílový obrázek morfologicky upravit. I jednoduchá úprava může vylepšit detekci pohybu jak je vidět z Obr. 2. Pokud se použije kromě eroze i uzavření a vhodné strukturní elementy je možné dosáhnout kvalitních výsledků viz. Obr. 7. To ale za cenu složitějšího nastavování pro konkrétní typ scény (dálnice, parkoviště,...). Naproti tomu použití některých méně obvyklých operací vedlo k více obecnému řešení (např. mediánový průměr, nebo funkce uzavření menších/středních děr)



**Obr. 7 Ukázka dobře použitých morfologických operací při získání pozadí**

Obrázek byl vytvořen pomocí kombinací několika morfologických operací a průměrování pozadí klouzavým průměrem.

Oproti detekci pohybu, při získání pozadí nevadí menší oblasti s nesprávně detekovaným pohybem (pohyb v nich ve skutečnosti nenastal), pozadí pouze nebude

aktualizován. Také při aspoň přibližně dobře nastaveném adaptačním koeficientu  $\alpha$  veškeré pohybující se objekty za krátký čas z pozadí zmizí.

Pro scénu typu chodba, kde se nepředpokládá její změna, byl vyzkoušen následující postup. Protože se získané pozadí nemůže u nikdy změnit, byl nastaven vysoký práh pro určení zda je na obrázku pohyb. To způsobilo velmi zašumění prahovaný rozdílový obrázek. Pozadí tak bylo získáváno velmi dlouho. Výhodou bylo získání velmi kvalitního pozadí kde bylo možné zaručit nepřítomnost pohyblivých objektů. Tento postup ale lze uplatnit jen pokud víme, že se pozadí nezmění.

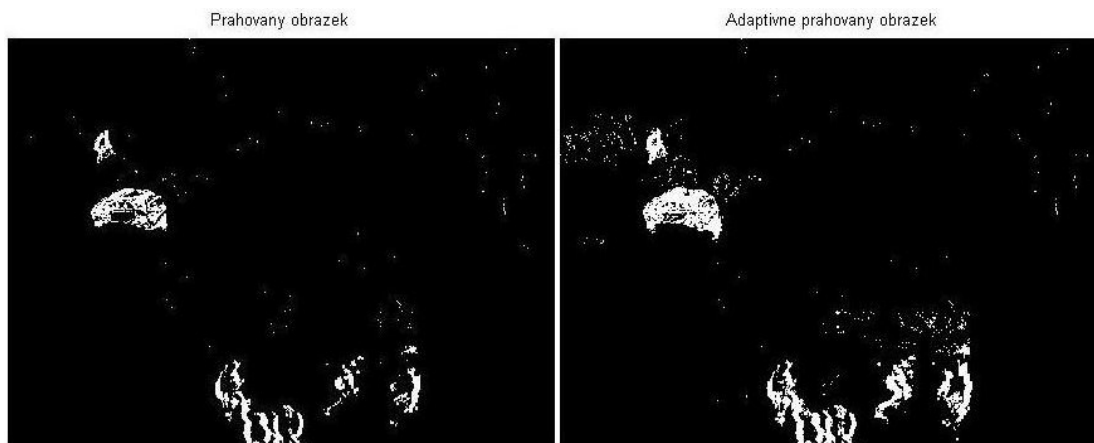
Při použití Running Gaussian average byly podle očekávané teorie dosahovány lepší výsledky (než použití obyčejného rozdílového snímku) i při obecnějším nastavení.

## **5.2 Post-processing popředí**

### **5.2.1 Lokálně proměnné prahování**

Při použití rozdílových metod detekce pozadí je získán rozdílový obrázek. Jeho prahováním se určí na kterých pixelech se vyskytuje pohyb. Výše v teorii popsáný problém je, jak nastavit tento práh. Pokud je nastaven příliš nízký objevuje se v popředí množství šumu. Když je nastaven vysoko nejsou pohybující se objekty v popředí úplné, případně „děravé“. Kompromis volby prahu lze obejít jeho lokálně proměnným nastavováním.

Práh je nastavován různě na různých částech obrázku, podle toho jestli se na nich vyskytl pohyb. Obrázek je nejprve rozdělen na 6x4 dílů. Dále byl hledán příznak pohybu v dané oblasti, který vykazuje co největší rozdílnost pro oblast kde se vyskytuje část pohyblivého objektu a oblast kde je pouze statické pozadí. Testem bylo zjištěno, že dostatečnou odlišnost má šířka histogramu. Histogram části rozdílového obrázku s výskytem pohybu je širší a obsahují víc nenulových prvků vyšších hodnot. Naproti tomu histogram části se statickým pozadím je velmi úzký, špičatý a většinu složek má umístěných vlevo. Odstraněním složek více vlevo a následným sečtením histogramu byl získán koeficient, který po normování docela dobře odděloval části obrázku s pozadím a části s pohybem. Podle jeho hodnoty byly vytvořeny 3 prahy, jeden pro části, ve kterých určitě nastav pohyb (vysoký práh), druhý pro oblasti kde pohyb určitě nenastal (nízký práh) a třetí pro části, u nichž nelze přesně rozhodnout (standardní práh). Porovnání proměnného prahování a prahování s jedním prahem ukazuje následující obrázek.



Obr. 8 Lokálně proměnné prahování

## 5.2.2 Zaplnění velkých děr

### Důvody vzniku děr

Binární obrázek popředí často obsahuje objekty s jednou nebo několika děrami nebo jsou detekovány pouze okolí hran pohybujících se objektů. Tyto díry mohou být tak velké, že je nelze zaplnit morfologickým uzavřením bez vedlejších efektů na tvaru objektu. To se stává nejčastěji, u velkých objektů s jednodušími plochami (např. velká dodávka, tramvaj), ale při pomalém pohybu se to může stát i na relativně menších objektech. Příklad je vidět na předchozím obrázku, kde dole jsou lidé vycházející z budovy. Je to způsobeno právě jednoduší plochou, kdy daný pixel má vždy tu stejnou hodnotu i přesto, že se objekt pohybuje. Jiným důvodem mohou být tmavě vypadající okna aut, které nemají dostatečný kontrast vůči tmavé silnici, na které jedou.

### Snížení snímkové rychlosti kamery

Jedna možnost jak se tomu vyhnout je použít každý druhý, případně  $x$ -tý snímek z videa. Kde  $x$  zvolíme podle toho jak pomalu (relativně pomalu) se objekty pohybují. Samozřejmě při velkém  $x$ , ztrácíme mnoho informací a nejedná se o příliš korektní řešení. Testy bylo ověřeno, že tento postup skutečně velmi dobře funguje a u pomalu pohybujících se objektů zlepšuje výsledek a to při takovém přeskakování snímků, které není ve videu rušivé. Pro rychle se pohybující objekty toto nelze použít, výpadek i jen několika málo snímků působí příliš rušivě.

### Funkce `imfill`

Na první pohled vhodným řešením se jeví použití funkce matlabovské funkce `imfill`. Toto řešení ve většině případů není příliš vhodné a v jiných ani nefunguje. Není vhodné v případě, kdy náhodná spojení objektů nebo šumu dávají vzniknout uzavřeným děrám. Ty jsou pak zaplněny zmíněnou funkcí. To se stává např. pokud jsou ve scéně stromy s pohybujícími se větvemi, nebo větší množství lidí v davu. Funkce navíc nefunguje pokud není díra uzavřená, stačí když v hranici chybí jediný pixel. Což

je naprosto přirozené, protože potom není podle čeho algoritmicky tuto polozavřenou díru detekovat. Částečně je možné toto omezit morfologickou dilatací, ale tím mohou vzniknout i nechtěné uzavřené díry. Navíc se vyskytl i případ kdy v binárním obrázku chyběla celá jedna hrana pohybujícího se objektu. V některých scénách, např. „dálnice“, kde nehrozí vznik falešných děr tento způsob lze s částečným úspěchem použít.

### **Detekce silulety**

Úspěšné řešení zaplnění uzavřených i neuzavřených děr vychází z „nasvícení objektu ze stran“ a vrhaného „stínu“. Na (binární) popředí je „posvíceno“ např. zleva. Všechny bílé pixely (označující objekt) vrhají stín a každý pixel, který se nachází v tomto vrženém stínu je také označen jako popředí. Tím je získána „silueta zleva“. Na původní obrázek je opět posvíceno z jiné strany a vržený stín připojen k objektu. Dostaneme siluetu zprava. To je opakováno pro všechny směry nahoru a dolů a pro 4 šikmé směry světla. Celkem dostaneme 8 siluet. Každá z nich dokáže zaplnit (i velmi silně neuzavřenou) díru z daného směru. Průnikem těchto siluet se zruší stín za objektem.

Získáme pouze objekt, který má zaplněno naprostou většinu děr a relativně dobře doplněné okraje objektu. Metoda nemá, na rozdíl od morfologie žádné nastavitelné parametry. To zvyšuje její obecnost. Nepoškozuje obrys objektu, na rozdíl od morfologie. Ta by musela být předimenzována pro uzavření srovnatelně velkých děr a poškodila by obrysy objektu.

Oproti získání ideálního tvaru pohybujícího se objektu metoda vykazuje menší nedostatky. V některých případech není průnik všech 8 siluet schopný eliminovat některé pixely, které objektu v popředí nenáležejí nebo detekovat všechny pixely, které mají k objektu patřit. Případně vznikají příliš uměle vypadající tvary. To je možné omezit použitím siluet z víc směrů než 8. Siluety z šikmých pohledů jsou vytvořeny rotací obrázku. Vzhledem k její výpočetní náročnosti se tím velmi prodlužuje doba zpracování jednoho obrázku

### **Aktivní kontury**

Velmi dobré, řešení je použití aktivních kontur, active appearance modelů, nebo podobných metod. V [3] v kapitole o analýze pohybu je tímto způsobem dosaženo ideálního tvaru pohybující se osoby na obrázku popředí. Případně je možné takto vykreslit kolem pohyblivých objektů hranici. Metoda nebyla vyzkoušena, z důvodu pozdního objevení možnosti jejího použití při detekci pozadí.

## **5.3 Mixture of Gaussian**

Model byl testován na nejvíce různých scénách. Na většině z nich fungovala relativně dobře. Některé nedostatky jsou principiálního rázu u většiny zatím

vyzkoušených i teoreticky prostudovaných algoritmů. Několikrát se vy průběhu vykreslování pozadí vyskytla chyba způsobící pád Matlabu. Opakovaný spouštěním se bylo pokoušeno chybu vyvolat a získat nějaký výpis před pádem Matlabu. Pokusy nevedly k úspěchu. Bylo zjištěno, že chyba souvisí pravděpodobně s přetížením systému, protože se vyskytovala více při maximálním počtu Gaussianů ( $K=7$ ) a při spuštění větším množství jiných programů. Dále bylo zjištěno, že pokud v pořádku proběhne první cyklus, chyba se už neobjeví do dalšího spuštění programu.

### **Počáteční nastavení modelu pozadí**

Metoda byla nejprve implementována pro  $K=3$  (pro 3 Gaussiany). Po otestování funkčnosti a dobrých výsledcích byla naprogramována pro téměř libovolný počet Gaussianů. Spodní hranicí je právě 3, protože použití dvou Gaussianů odpovídá jedno popředí a jedno pozadí. Pro statické pozadí nemá velký význam takto složitý algoritmus a je doporučeno použít Running Gaussian average. Shora je  $K$  omezeno na 7. Při použití většího počtu už není udáváno velké zlepšení, [10] [8] což bylo prakticky ověřeno. Jediná možnost smysluplného použití tohoto algoritmus pro pozadí bez pohybu, je při pohybu kamery, např. kvůli otřesům nebo větru. Pixely na hranách objektů by pak představovali proměnné pozadí, na které by se tento algoritmus adaptoval. Experimentálně bylo ověřeno, že i při 7 Gaussianech už není viditelné příliš velké zlepšení oproti nižšímu počtu. Navíc z výsledků bylo vidět, že do modelování pozadí se stejně nezapojuje všech 7. V [18] se uvádí, že pro srovnání s vlastním programem použily MoG  $K=10$ , kde desíti Gaussiny bylo ale modelováno jen „velmi málo pixelů“. Při  $K=5-7$  a příliš pomalu (relativně pomalu k nastavování obnovovací rychlosti) se pohybujících objektech ve scéně bylo velké riziko naučení pohybujících se objektů do pozadí. To se nestávalo při modelování 5 Gaussiany.

Obvykle je výhodné zvolit větší počet Gaussianů než 3. Pro jednoduché scény (menší počet různých pozadí) je to naopak méně vhodné. Jednotlivé Gaussiany si mezi sebou rozdělí váhy (jejichž součet se musí rovnat jedné) a potom déle trvá přepínání mezi jednotlivými pozadími. Proto by mohlo být dalším vylepšením adaptivně měnit počet Gaussianů podle toho jak moc jsou využívány. Díky tomu je také vhodné pro větší počet Gaussianů zvolit nižší práh  $T$  (v rovnici (8)). Pro  $K=3$  byl zvolen práh  $T=0,7$ , pro  $K=5$  je vhodnější práh  $T=0,6$  nebo i méně. Nebyli zjištěny problémy ani při snížení prahu na 0,4.

Směrodatné odchylky  $\sigma$  byly a váhy  $w$  byly inicializovány podle [10]. Střední hodnoty byly inicializovány rovnoměrně přes celý interval 0-255. Pozadí lze také zvolit na základě mediánu z více snímků. Na to je ale potřeba několik snímků a ty mohou být využity přímo v MoGu inicializovaného na libovolnou hodnotu pozadí. [10]

## Možnosti urychlení

Zpracování jednoho snímku i pro nastavení  $K=3$  trvá neúměrně dlouho a rozhodně by nebylo možné jej takto použít pro real time sledování. Pro představu je uvedena následující tabulka.

K	3	4	5	6	7
čas[s]	2,44	3,07	3,79	4,60	5,66

Tabulka 1 Doba zpracování jednoho snímku MoG

Hodnota udává čas na úplné zpracování jednoho snímku (1 cyklus hlavní smyčky). Byla vypočtena jako průměr z dostatečného počtu snímků, aby zůstávala konstantní. Bylo zapnuto zobrazování všech výstupních obrázků tj, aktuální snímek, rozložení Gaussianů, popředí, pozadí a buffer. Hodnoty tvoří v grafu lineární funkci.

Výrazné urychlení by mohlo nastat při převedení celého kódu do jazyku C. Pokud bychom přesto požadovali ponechání programu v Matlabu bylo by možné program urychlit vynecháním výpočtu odmocniny a upravit kód na místech kde s výpočtem odmocniny počítá. [10] Nebo výpočet nahradit některou z aproximací odmocniny. Další zrychlení by bylo dosažení prostým snížením  $K$  (počtu Gaussianů). Nastavení  $K=7$  nedává už o moc lepší výsledky než pro  $K=5$ , které by nebylo možné dotáhnout na odpovídající úroveň např. morfologickými operacemi. Což by bylo pravděpodobně rychlejší. Pokud by to nestačilo je možné výpočet paralelizovat např. pomocí Parallel Computing Toolboxu. Nezávisle vypočítávat by byly možné

1. jednotlivé pixely – Parametry Gaussianů pro jednotlivé pixely na sobě vůbec nezávisí, proto by bylo nejjednodušší obrázek rozdělit na tolik částí, kolik máme k dispozici výpočetních jednotek. Jediná závislost pixelů vzniká při morfologických operacích. Obrázek nesmí být rozdělen na menší části než je největší morfologický operátor. Jednotlivé části by měly být mírně překryty. Další problém by mohl nastat při převodu 3D matice na 2D matice v nejnižší funkci, kdy by se muselo ošetřit předávání správných pixelů. Pak by bylo nutné pouze zařídit aby výpočet jednotlivých částí obrázků netrval některému z jader procesoru déle než jinému, nebo by bylo nutné na nejpomalejší čekat a zobrazovat spojený výsledek ze všech částí až po obdržení všech dílčích výsledků.
2. parametry jednotlivých Gaussianů – Je možné paralelizovat většinu výpočtů kde se updatují parametry (podle rovnic (5)-(7)) a každý pro každý Gaussian počítat změnu parametrů zvlášť. Nicméně to naráží na několik řádků kódu, které pracují se všemi Gaussiany zároveň. K nim patří např. řádek 104 pro normování vah na  $\text{suma} = 1$  (sčítá všechny váhy) nebo všechny řádky pro seřazování funkcí sort, 3D výpočet „match“ na řádce 73, atd. Tato paralelizace není příliš výhodná, protože úseky kódu k paralelizaci jsou příliš často přerušovány jedním řádkem, který musí mít všechny hodnoty zároveň.



3. úseky kódu – Protože velké množství výpočetního času zabere zobrazování (při testování pomocí profileru byly zobrazovací funkce (imshow) na prvních místech, spolu s výpočtem odmocniny). Výpočet a zobrazení může být odděleno. Ideální řešení by pak bylo přepsat kód tak, aby zobrazovací úseky byly na konci cyklu pro zpracování jednoho snímku.

Podobná paralelizace by byla možná i při použití jazyka C, kdy náhodou výpočetní rychlost nestačila. Což se může stát např. při snímání videa s vysokým rozlišením.

### Poznámky k výsledkům, ukázkové obrázky

Kvalita výsledků byla ovlivněna vhodným nastavením adaptačních parametrů. V původním článku [9] se uvádí pouze jeden adaptační parametr, ale bylo zjištěno, že pro zlepšení je možné nastavit rychlosti adaptace pro všechny tři parametry Gaussianu zvlášť. Tím je možné přizpůsobit model pozadí konkrétní scéně (např. předpokládaná rychlost pohybu ve scéně, velikost objektů,...). Následující obrázek ukazuje jedno z výsledných pozadí. Větve na spodním okraji obrázku se pohybují, ale přesto jsou zařazeny do pozadí. Z popředí nejsou vynechány úplně, kvůli relativně velmi rychlému a relativně i náhodnému pohybu, kdy se Gaussiany nestihnou adaptovat na nové pozadí.



Obr. 9 MoG model, zleva aktuální obrázek, popředí a pozadí

### Rozšíření

Urychlené učení - Čím větší byla hodnota  $K$ , tím rychleji se model naučil. Např. na naučení scény „příroda“ z předchozí obrázku pro  $K=3$  bylo potřeba 85 snímků, pro  $K=5$  bylo jen 40 snímků a pro  $K=7$  byl naučený po 28 snímcích. To byla stále relativně vysoká hodnota, proto byl koeficient rychlosti adaptace  $\alpha$  na prvních  $x$  snímků nastaven jako  $1,5x \alpha$ . [10]. Model pro  $K=5$  byl potom naučen už po 20 snímcích. Pro každé  $K$  byl nastaven odlišný počet snímků pro zrychlenou adaptaci.

Filtrace šumu - Na tmavých místech obrázku (tmavý chodník, stín pod mostem) bylo popředí zatíženo šumem. Tomu se nelze vyhnout, protože při nízkém jasu je poměr signál šum větší a šum se projevuje v popředí. Je možné filtrovat na šum celý obrázek, ale ztrácíme ostrost pozadí. Bylo vyzkoušeno vybrat z obrázku pouze tmavé plochy, filtrovat je a vyfiltrované je vložit zpět do původního obrázku. Míra šumu se snížila. Nevýhodou byly artefakty na přechodu mezi filtrovanou tmavou plochou a zbytkem obrázku. Tyto artefakty se ale příliš neprojevovali v pozadí. Větší nevýhodou bylo,

pokud míra šumu překročila určitou hranici, nad kterou filtr nestačil. V obrázku popředí se potom objevily plošné artefakty zhoršující obrázek mnohem více než původní šum. Filtrace tmavých ploch proto nebyla ve výsledném programu použita. Místo ní byl na obrázek popředí použit mediánový filtr.

Buffer popředí – Na popředí byly použity standardní morfologické operace a některé další nestandardní. Dále byl vytvořen jeho buffer. Do něj se ukládalo popředí a zůstávalo v něm ještě po „x“ snímcích. Pozadí se potom obnovovalo, až když daný pixel po „x“ snímků nebyl označen jako popředí. Tím bylo zlepšeno výsledné pozadí. Použití bufferu se ale hodí jen pro scény, kde pravděpodobně nebudou žádné objekty zastavovat, nebo se rozjíždět, ale vždy jen projedou, např. dálnice. Jinak se zpomalí adaptace pozadí na tyto zastavující objekty.

Detekce pohybu v pozadí – Bufferu bylo dále pro lokálně proměnný koeficient rychlosti adaptace  $\alpha$ . Na místech, kde je přítomný velmi rychlý pohyb v pozadí, na který by model nebyl schopný se naučit je lokálně koeficient  $\alpha$  zvýšen. Tyto místa jsou detekována pomocí bufferu. Daný pixel musí být opakovaně x-krát po sobě zařazen mezi popředí (~pozadí, které ještě nebylo naučeno). Tímto zvýšením se nestává, že by se některá část objektů v popředí dostala do pozadí, protože objekt v popředí je obvykle v pohybu.

### **Závěr MoG**

Použitím MoG je možné relativně kvalitně detekovat pozadí i popředí. Nepřesnosti vychází hlavně s nevhodně nastavených parametrů nebo použití nevhodných vylepšení vzhledem ke scéně. Při alespoň zběžné znalosti scény dává po vhodném nastavení dobré výsledky.

## **5.4 Detekce pohybu v barevném videu**

Při testování barevného videa byly vyzkoušeny různé možnosti odečítání pozadí. Pokud byla rozdílnost barev (ať už byla zjišťována v jakémkoliv barevném prostoru jakoukoliv metodou) větší než nastavený práh byl pixel detekován jako popředí. Byl otestován tyto kombinace

1. RGB model a euklidovské vzdálenost nezávisle v jednotlivých barevných složkách, výsledné popředí tvořil jejich průnik
2. RGB model a klasická euklidovská vzdálenost
3. RGB model a měření úhlu vektorů
4. Lab model a euklidovská vzdálenost
5. XYZ model a euklidovská vzdálenost
6. Luv model a euklidovská vzdálenost
7. HSV model a euklidovská vzdálenost

## 8. kombinace HSV a RGB, vysvětlená později

Pro převod mezi výchozím RGB ostatními modely byly použity standardní funkce Matlabu

```
I_hsv=rgb2hsv(Iakt);
```

Případně pro modely bez přímé konverze

```
cform=makecform('srgb2lab');  
I_Lab=lab2double(applycform(Iakt,cform));
```

Pro všechny modely byly nastaveny stejné výchozí podmínky a následné zpracování. Výsledky prvních šesti modelů (ve srovnání s šedotónovým modelem byly většinou horší. Byly porovnávány kvalita detekce popředí a velikost šumu. Aby se vyloučil vliv toho, že některá scéna by mohla být vhodnější pro daný barevný model bylo testováno 3 různé scény. Scéna „parkoviště“, scéna „dálnice“ a scéna „ulice“. Podle očekávání [19] byl nejhorší výsledek pro XYZ model. Vykazoval špatnou detekci a téměř konstantní šum přes celý obrázek, nejen na hranách objektů. Většina dalších modelů měla srovnatelnou kvalitu detekce popředí i podobnou přítomnost šumu.

Např. model 3. potlačoval šum na hranách objektů, ale nebyl schopný detekovat málo barevná auta. Selhal např. při detekci bílého auta. Musí ale být dodáno, že auto jelo na šedé vozovce. Bílá a šedá mohou mít podobný barevný tón a lišit se pouze intenzitou jasu a na invariantnost pro změnu osvětlení je tento model vytvořen. Model také samozřejmě výborně ignoroval stíny vrhané objekty.

Po srovnání všech modelů vyšel pravděpodobně nejlépe, klasický model RGB a euklidovská vzdálenost. Relativně dobře detekoval popředí, ale přidával více šumu ve srovnání s šedotónovým videem.

Lepší výsledky byly také očekávány u HSV modelu. Na některých obrázcích měl model dobré výsledky, co se týká kvality detekce i šumu. Při použití pouze složky H byl model naprosto jasově invariantní a výborně ignoroval stíny. Při použití složky H a S invariantnost na jas mírně poklesla. Samostatným testem, bylo zjištěno, že při přesvětlení obrázku má složka S nižší hodnoty, což odpovídá lidskému vidění, při velkých intenzitách světla oko hůře vnímá barvy. Větším problémem než nižší invariantnost na změnu jasu se ukázala být periodicitu složky H modelu. V modelu je červená barva vyjádřena čísly blízkými nule (zhruba 0-0,15) a čísly blízkými jedné (0,85-1). Pokud jsou od sebe obrázky odečteny, na plochách s červenou barvou v obou obrázcích by měly výsledné hodnoty být blízké nule. Pokud se ale na jednom z obrázků vyskytuje „vysoká červená“ a na druhém „nízká červená“ hodnoty jsou naopak vysoké. Tyto plochy jsou pak algoritmem označeny jako popředí. Následující obrázek ukazuje výsledný prahovaný rozdílový obrázek.

lbw6



**Obr. 10** Vliv periodičnosti HSV modelu na rozdílový obrázek  
Jedná se o tento obrázek z parkoviště; obě dodávky stojí.

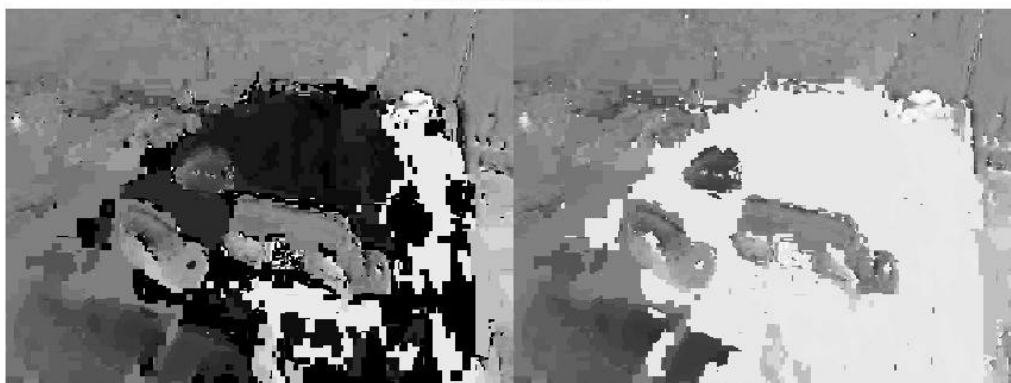


**Obr. 11** Výchozí obrázek k ukázce vlivu periodičnosti HSV modelu

Bylo navrženo řešení, několik řešení. Nejlépe fungovalo toto: přičíst ke všem červeným hodnotám 0,91 (konstanta stanovená tak aby „nízké červené“ byly přesunuty na hodnoty „vysokých červených“). Poté bylo na místech, kde byly původně „vysoké červené“ odečteno 0,91. Tím se oba druhy červených dostali na podobné hodnoty. Následující obrázek ukazuje složku H (barevný tón) před úpravou a po úpravě, kdy

všechny plochy s červeným odstínem barvy jsou mají blízké hodnoty. Tmavé plochy na levém obrázku označují místa s „nízkou červenou“ a bílé s „vysokou červenou“

Původní a upravená složka H



**Obr. 12 Odstranění periodičnosti HSV modelu**

I když na původním obrázku není vidět červená, ale šedá barva. V modelu HSV barevný tón těchto ploch odpovídá červené barvě. Nedostatek této metody, je že chybu periodicity pouze přesouvá do jiné méně obvyklé barvy. Chyba tak není ve výsledku příliš výrazná. Mohou se objevit scény, kdy se chyba projeví víc.

Dalším vyzkoušeným řešením tohoto problému byla výše zmíněná kombinace HSV a RGB modelu. Na všech místech v obrázku byl použit HSV model. Pouze na místech kde, kde hrozil výskyt chyby z periodicity, byl použit model RGB. Místa byla detekována jako hodnoty blízké nule nebo jedničce. Metoda dobře vyřešila problém s periodicitou. Současně ale přestala být invariantní na změnu osvětlení na místech, kde byl HSV model nahrazen RGB. Toto nahrazení proběhlo většinou v částech obrázku kde byly zrovna očekávány stíny (např. vozovka dálnice). Na místech kam se stín z pohybujících se objektů dostat nemohl k nahrazení nedošlo. Tím se prostorově omezila invariantnost na změnu jasu. Částečná invariantnost zůstala, ale nebyla dostatečně vysoká, aby to bylo důvodem použít tuto kombinaci modelů.

Závěrem k barevným modelům je možné říct, že očekávané vylepšení, nebylo dosaženo v dostatečné míře. Kvůli obecnosti a použitelnosti i pro nebarevné kamery byly další algoritmy vytvářeny pro šedotónové video. Další výhodou při použití šedotónového videa je výrazně rychlejší výpočet. Zpracovává se jeden jasový kanál, místo 3 barevných. Po úplném odstranění chyby periodicity by měl být model HSV lepší než šedotónový snímek. Pro některé pokročilejší algoritmy detekce pozadí, by barva mohla být přínosnější. Kvůli obecnosti, ale nebyla vyzkoušena.

## 5.5 Použití významných bodů

### 5.5.1 Moravcův operátor

Vzhledem k jeho jednoduchosti i hodnocení různých autorů, byl operátor naprogramován pouze jako test. V testovací obrázku (umělé vytvořené geometrické obrazce a různé šikmé hrany) operátor našel rohy, ale některé celkem nepřesně. Na reálném obrázku operátor nevyhledal ani zjevné rohy a díky šumu navíc označil za roh i spoustu naprosto falešných rohů. Případné výsledky jsou v přílohách.

### 5.5.2 Harrisův detektor

Při tvorbě Harrisova detektoru jsem po schválení vedoucím práce použil pro inspiraci kód z [12]. Předpokládal jsem, že se bude jednat o mnohem rychlejší implementaci než bych vytvořil sám. Kód jsem si dále upravil (hlavně odstranění zbytečně vícenásobně vnořených funkcí a přepsání názvů proměnných. Detektor dával relativně dobré výsledky, při vhodném nastavení. Nejvíc záleželo na počáteční filtraci obrázku. Dosažené nastavení fungovalo dobře obecně pro většinu scén.

Některé významné body, jsou i přes „filtraci okrajů“ nalezeny na okraji obrázku. Tyto body zbývá jen odstranit.

Bylo naprogramováno vyhledávání konkrétního počtu nejlepších významných bodů. Kvůli nutnému odstranění bodů z okrajů obrázku nelze vyhledat přesný počet nejlepších bodů, ale vždy bude vyhledáno o něco méně bodů (okolo 5-10% bodů). To příliš nevadilo, protože kolik bude vybráno nejlepších bodů bylo zvoleno odhadem.

Různé modifikace Harrisova operátoru (popisované rovnicemi (33)-(36) nepřinesly přílišné vylepšení. Dvě z modifikací fungovali spíše jako detektory šikmých hran. (což může někdy být vhodné, jak bylo v jedné scéně zjištěno). Zbývající modifikace nepřinesla výrazné vylepšení. Pro konkrétní určení lze některé modifikace využít, ale jinak jsem nezjistil jejich větší přínos. Následující obrázek ukazuje výsledek původní nemodifikované verze Harrisova detektoru.

Obrázek s 92 významnými body způsob 1



**Obr. 13 Harrisův rohový detektor**

Zabránění nahromadění významných bodů na jednom místě bylo nakonec vyřešeno rozdělením obrázku na  $M \times N$  volitelných částí. V každé byl počítán počet významných bodů a pokud jejich počet překročil hodnotu odvozenou z počtu částí nebyly další body přidávány. Takto vzniklo rovnoměrné rozložení bodů po obrázku (s přihlédnutím ke skutečné poloze objektů ve scéně; na velkých plochách s konstantním jasnem žádné rohy nebyly uměle přidány)

Harrisův rohový detektor byl také naprogramován jako základ pro afině invariantní Harrisův detektor.

### **5.5.3 Metody sesouhlasení z významných bodů**

Byly vyzkoušeny 3 možnosti sesouhlasení. Nejjednodušší metoda (vyhledání bodů podle nejbližší vzdálenosti) i přes jednoduchost fungovala správně. Některé významné body byly sice vyhledány na pohybujícím se objektu. Výsledné obrázky byly ale sesouhlaseny správně (korekce posunu byla vypočtena jako nejčastější hodnota ze všech posunů významných bodů). I druhá metoda pracovala správně. Drobný nedostatek je horší přesnost korespondencí bodů, díky postupnému vymazávání významných bodů z obrázku 2. Chyba byla opět kompenzována výběrem nejčastější hodnoty posuvu (jako v prvním případě). Třetí metoda také úspěšně sesouhlasila vyzkoušené obrázky. Její funkce je vidět z následujících dvou obrázků. Pokud jsou rohové body příliš blízko sebe, nejsou v dalších výpočtech použity. Na prvním obrázku je nastavená malá minimální blízkost bodů. Na druhém větší, což se projevilo snížením počtu nalezených korespondencí.





**Obr. 14 Korespondence rohových bodů při malé minimální blízkosti bodů**



**Obr. 15 Korespondence rohových bodů při větší minimální blízkosti bodů**

Tímto způsobem lze také ošetřit, když je vyhledáno velké množství rohových bodů. Použijí se pouze body dostatečně vzdálené od sebe. Některé další výsledky jsou opět v přílohách.

Na zkoušku byla naprogramována i metoda RANSAC. Bylo zjištěno, že pro tuto metodu se hodí velké množství významných bodů (100+ i víc), tak aby co nejvíc bodů bylo umístěno ve statickém pozadí. Dále bylo zásadní nastavení požadované přesnosti. Při dostatečné přesnosti jsou výsledky vždy správné, při nižší (a při malém počtu významných bodů) jsou skoro pokaždé špatné, viz obrázky v přílohách. Velkou nevýhodou je pomalost výpočtu „ohodnocení přesnosti zvoleného posuvu“. To je způsobeno množstvím významných bodů, u kterých musí být vypočtena vzdálenost. Pokud by se optimalizovala tato část jednalo by také o velmi rychlý algoritmus, protože požadovaná přesnost je často dosažena za relativně málo iterací.



## 5.6 Korelační metody

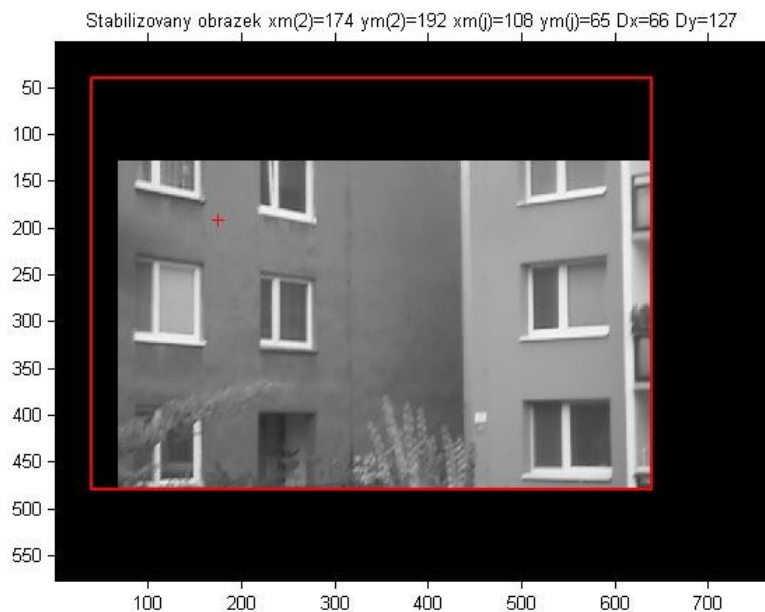
### 5.6.1 Optický stabilizátor pomocí korelace

Metoda vykazovala dobré výsledky. Na její základě byl navržen optický stabilizátor. Při vhodném výběru objektu byl obraz stabilizovaný. Pouze občas byly vlivem šumu viditelné 1pixelové posuny obrázky. Při nevhodném výběru objektu (např. opakující se struktury) obraz velmi často „odskakoval“.

Před vlastní stabilizací je uživatel vyzván k výběru objektu, podle kterého se bude stabilizovat. Následující obrázky ukazuje jak program funguje. Na prvním je fungující stabilizace (bylo vybráno číslo popisné), na druhém je špatná stabilizace (byl vybrán roh spodního okna (ne horního, které je na obrázku)). To potvrzuje nutnost výběru vhodného objektu. Čísla za nadpisem udávají referenčního objektu, současného objektu a vypočtený posun.



Obr. 16 Ukázka optického stabilizátoru na základě korelace



**Obr. 17 Špatná stabilizace vlivem nevhodně vybraného objektu**

Toto „odskakování“ bylo částečně potlačeno váhováním Gaussovým filtrem, umístěným do polohy poslední známé správné polohy šablony. Pokud nedojde k „výpadku sesouhlasení“ (vysoký rozdíl po sobě následujících poloh šablony) je filtr relativně úzký. Při výpadku sesouhlasení se na použit širší filtr, aby mohla být nalezena správná poloha šablony.

## 5.6.2 Normalizovaná korelace a automatické vyhledání šablony

Pro své lepší vlastnosti byla vyzkoušena i normalizovaná korelace. Pro větší snímky nebo šablony je v přímém provedení (podle vzorce(20)) nepoužitelná, kvůli nízké rychlosti. Měla být naprogramována normalizovaná korelace podle rovnic popsanych v teorii. Předtím bylo ale zjištěno, že se rychlá verze normalizované korelace je jedna z vložených funkcí Matlabu. Rychlá verze netrpí numerickou nestabilitou popsanou v [22], což bylo experimentálně ověřeno.

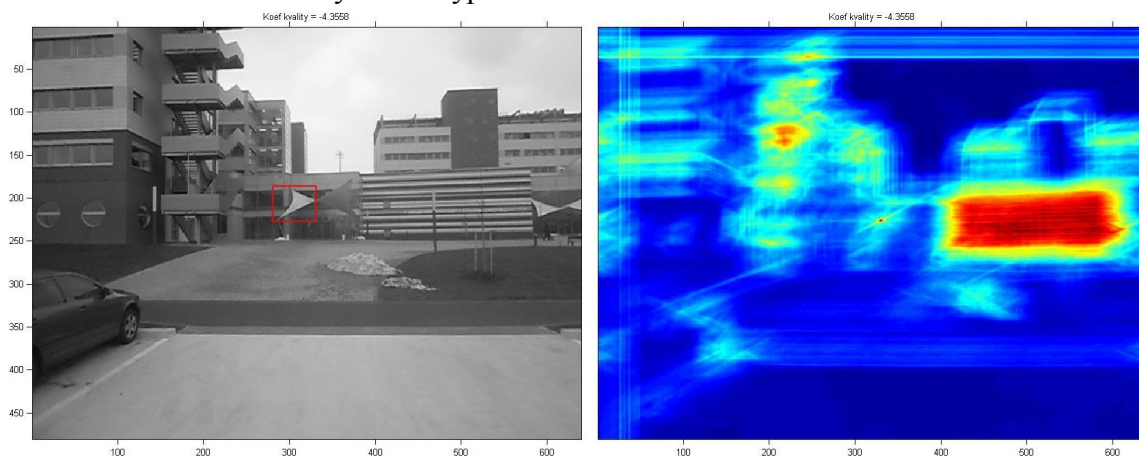
Normalizovaná korelace byla dále využita při automatickém vyhledání šablony pro template matching a s tím souvisejícím hodnocení kvality šablony.

Normalizovaná korelace byla schopná pracovat i na obrázcích bez zvýraznění hran. Na těchto obrázcích vykazovala menší přesnost (chyba byla asi 16,6%). Po zvýraznění hran byla přesnost stoprocentní (pro správně vybranou šablonu).

### Hodnocení kvality šablony

Při pokusech bylo zjištěno, že i šablona, která je hodně odlišná od okolí není pomocí korelací vyhledána spolehlivě. Pro ideální šablonu vypadá výsledek (normované)

korelace jako jedno úzké maximum a nízké hodnoty na zbytku obrázku. Korelační obrázek nevhodné šablony může vypadat i takto:



**Obr. 18 Korelační obrázek na první pohled vhodné šablony**

Šablona vizuálně vypadá velmi odlišná, výsledek korelace je ale jiný. Stabilizace kamery podle této šablony samozřejmě hned na druhém snímku selhala. Proto bylo vymyšleno hodnocení kvality šablony.

Pro vybranou šablonu byla provedena korelace. Koeficient kvality byl vyjádřen jako

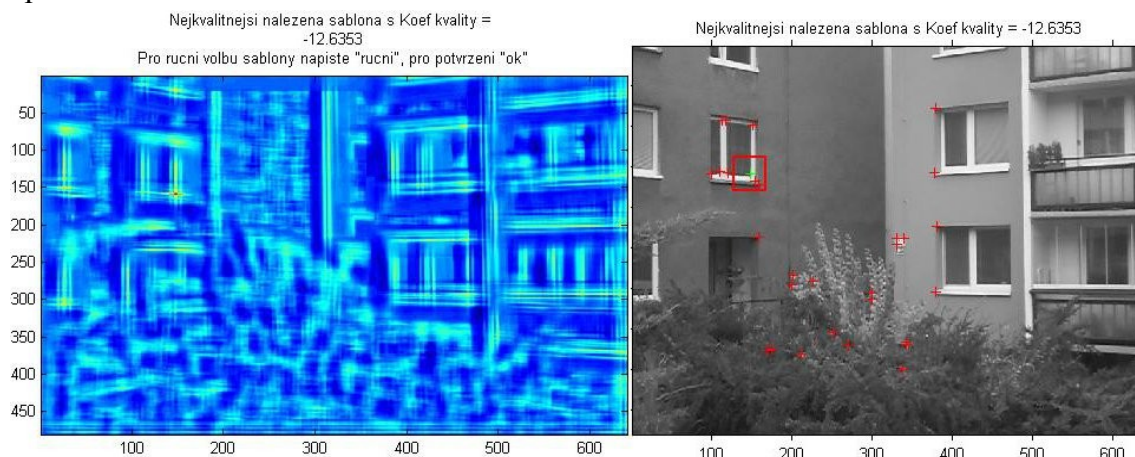
$$koef\_kvality = \log\left(1 - \frac{x}{p}\right),$$

kde  $x$  značí počet pixelů s hodnotou korelace nižší než 0,9 maximální hodnoty a  $p$  počet všech pixelů. Výsledek byl logaritmován, kvůli zvýraznění rozdílů při malých hodnotách koeficientu kvality. [36] Nejlepším šablonám odpovídá  $koef\_kvality$  okolo -15, pro stále velmi dobré šablony vychází  $koef\_kvality$  od -15 do -10. Pro hodnoty -10 až -8 je šablona už špatná a bývá vyhledána v obrázku správně jen díky váhování Gaussovskou funkcí a to ještě ne pokaždé. Pro vyšší hodnoty než -8 doporučuji vybrat novou šablonu. Hodnocení kvality šablony je použito při automatickém vyhledání šablony, ale může být použito na ověření kvality vybrané šablony.

### **Automatické vyhledání šablony**

Na obrázku kde má být šablona nalezena jsou rovnoměrně detekovány významné body. Body blízko okrajů jsou odstraněny. Pro všechny zbylé významné body je vytvořena šablona se středem v tomto bodě a volitelnou velikostí. Šablony jsou ohodnoceny  $koef\_kvality$  a ta nejlepší je vybrána. Pokud ani ta nejlepší není dostatečně kvalitní je navržen ruční výběr šablony. Ten je navržen i pro dostačující nejlepší šablonu, v případě zájmu uživatele. Automatické vyhledání šablony může být použito pro zautomatizování optického stabilizátoru stabilní kamery. Následující obrázek ukazuje výslednou nejlepší šablonu a její korelační obrázek. Program byl dále vyzkoušen na několika dalších videích. Byly vyhledávány kvalitní šablony, jenže z větší části na pohybujících se objektech. To ovšem algoritmus nemohl vědět. Eliminovat

body na pohybujících se objektech je obtížné. Kvůli otřesům kamery standardní metody detekce pohybu získají většinou výsledný pohyb i na hranách objektů, které jsou vhodné pro výběr šablony. Možností je trasovat vybrané rohové body na několika prvních snímcích videa. Body, které budou vykazovat velký posun budou zamítnuty pro další zpracování.



Obr. 19 Výsledek automatického vyhledání šablony

### 5.6.3 Otáčející se kamera

Program je rozdělen na dvě logické části – učení šablon, kdy je uživateli opakovaně předkládán výběr šablony a samotná detekce pozadí, kdy je pomocí naučených šablon vyhledáván správný obrázek z první otočky. Po naučení šablon je vhodné je uložit do souboru. Na detekci pozadí byl použit Running Gaussian average. Použití MoG by nebylo možné kvůli delšímu učení a omezené velikosti videa umístitelného do Matlabu. V praxi by samozřejmě bylo možné použít i MoG, jen by počáteční naučení by trvalo odpovídající dobu. Při výběru vhodných šablon se jako velmi vhodné ukázaly být horní okraje střech nízkých domů, pokud je šablona dostatečné velikosti. Pak je v šabloně jedna výrazná hrana a obvykle „prázdný prostor“. Tomu pak nejlépe odpovídá jen správná pozice šablony. U takto nahoře umístěných šablon navíc nehrozilo jejich zakrytí projíždějícím autem. Dále bylo zjištěno, že dostatečně velká šablona bývá vyhledána skoro vždy správně.

I přes sesouhlasení bylo zjištěno zkreslení a ani sesouhlasení nebylo přesné na 1 pixel. V popředí se proto objevují zvýrazněné hrany objektů, vliv má i posun předchozího a aktuálního obrázku pozadí. Na druhou stranu pohybující se objekty jsou detekovány velmi dobře. To je způsobeno tím, že rozdílový obrázek je tvořen ze dvou různých otoček kamery a ne ze dvou po sobě následujících snímků. Potom nevznikají žádné problémy s děrami v objektu ani žádné jiné podobné problémy. Zkreslení by bylo možné omezit otáčení kamery kolem osy veden snímacím čipem fotoaparátu.

Metodou popsanou v teorii (pomocí lícovacích bodů) byla vyzkoušena korekce geometrického zkreslení. I přes relativní úspěch nebyla korekce dostatečně dokonalá, tak aby bylo možné natáčet bez stativu. Po použití stativu se zkreslení zmenšilo, ale přesto se ve výsledku stále projevuje.

Místo ručního výběru šablon by mohl být použit automatický, ale některé šablony by mohli být nalezeny nevhodně nízko u vozovky, případně na pohyblivé objektu. Ty by potom mohli být při některé otočce zakryty projíždějícím autem.

## 6 ZÁVĚR

Práce se zabývala různými metodami na získání statického pozadí scény, případně detekci pohybu, která s tímto úkolem souvisí. Byly zmíněny typy scén, pro které jsou nejčastěji výsledné programy používány a vytvořena krátká databáze jejich videí.

Teoreticky bylo nastudováno více algoritmů pro detekci pozadí. Každý měl jiné vlastnosti a různé výhody a nevýhody. Další nedostatky byly také často objeveny při implementaci a vlastním testování. Např. problém s periodicitou barevného modelu HSV. Pro odstranění nebo alespoň potlačení nevýhod byly navrženy různé metody. Kromě toho bylo testováno nastavení parametrů jednotlivých algoritmů, pro získání co neideálnějšího pozadí, buď obecně nebo alespoň pro daný typ scény. Výsledky většiny metod byl relativně závislé na jejich vhodném nastavení. Po určitém experimentování bylo možné odhadnout sadu parametrů, které dávají optimální výsledky. Pak měla většina metod dobré výsledky, někdy ale jen na daném typu scény. Např. velmi dobré pozadí bylo získáno Temporal median filterem, jehož jediným parametrem byla délka buferu.

Jako jeden z velkých problémů byl objeven překvapivě velký vliv otřesů kamery. Bez těchto otřesů vykazovali i jednoduché algoritmy velmi dobré výsledky. I při umístění na statickém, stabilním podstavci bylo občas ve videu vidět efekt chvění na hranách objektů. Proto bylo navržena optická stabilizace. Pro kvalitní šablonu byla stabilizace vynikající i při přehnaně velkém pohybu kamery. Vlivem šumu, byla ale její přesnost někdy nižší, což se projevilo jako zmíněný pohyb na hranách statických objektů. Stabilizačního algoritmu bylo dále využito v programu pro otočnou kameru.

Pokud by měla být zvolena nejlepší metoda, je těžké vybrat nějakou pro obecné použití. Pokud máme k dispozici dostatek výpočetního výkonu lze použít MoG s optickou stabilizací a na post-processing aktivní kontury. Pokud nemůžeme použít takto náročnou kombinaci musíme kameru stabilizovat mechanicky a spokojit se s Running Gaussian average, který je oproti MoG velmi rychlý. Výběr konkrétní metody získání pozadí závisí na dané aplikaci, výpočetních možnostech a požadované přesnosti.

Další zlepšení výsledků by bylo možné např. použitím trasování objektů a testování jejich reálného tvaru a pohybu. Tím by se vyloučily falešné detekce vnikající na hranách. Případně lze využít zmíněné aktivní kontury nebo jiné optimalizační algoritmy. V původním plánu bylo i zpracování videa z jedoucího auta. Díky složité teorii, afině invariantního detektoru významných bodů to nebylo realizováno.

# Literatura

- [1] Svoboda, T., *Motion segmentation—separating background and foreground* [online]. Dostupné na <http://cw.felk.cvut.cz/lib/exe/fetch.php/courses/a4m33mpv/motionsegm.pdf>, citováno 2.5. 2012
- [2] “*Detection of potentially dangerous situations involving crowds using image processing*”, Intelligent Industrial Automation, [online] IIA99 (special session on vision-based intelligent systems for surveillance and traffic control), June 1 – 4, 1999, Genoa, Italy. Dostupné na: [dilnxsrv.king.ac.uk/papers/iaa99.pdf](http://dilnxsrv.king.ac.uk/papers/iaa99.pdf) citováno 2.5. 2012
- [3] Hlaváč V., Šonka M.: *Počítačové vidění*, Grada, Praha 1992, ISBN 80-85424-67-3
- [4] Senior, A., Feris, R., Tian, Y., *Class 2: Moving Object Detection*, [online], Video Surveillance E6998-007, Feb. 4, 2008, Dostupné na: [http://www.andrewsenior.com/technical/surveillanceclass/02\\_MovingObjectDetection\\_part1.pdf](http://www.andrewsenior.com/technical/surveillanceclass/02_MovingObjectDetection_part1.pdf), citováno 2.5. 2012
- [5] Cucchiara, R., Grana, C., et al. *Detecting Moving Objects, Ghosts and Shadows in Video Streams*, [online], Dostupné na: [cvrr.ucsd.edu/aton/publications/pdftpapers/96\\_cucchiara\\_r.PDF](http://cvrr.ucsd.edu/aton/publications/pdftpapers/96_cucchiara_r.PDF), citováno 2.5. 2012
- [6] Paul L. Rosin, Tim Ellis, *Image difference threshold strategies and shadow detection*, [online], Dostupné na: <http://denizen.cs.cf.ac.uk/Paul.Rosin/resources/papers/shadows.pdf> citováno 2.5. 2012
- [7] Z. Tang, Z. Miao, Y. Wan, *Background subtraction using running gaussian average and frame difference*, [online], Dostupné na: <http://www.springerlink.com/content/978-3-540-74872-4/#section=369101&page=1&locus=0>, citováno 2.5. 2012
- [8] Piccardi, M., *Background subtraction techniques: a review*, University of Technology, Sydney, [online], Dostupné na: <http://www-staff.it.uts.edu.au/~massimo/BackgroundSubtractionReview-Piccardi.pdf> citováno 3.5. 2012

- [9] Stauffer, Ch., Grimson, W., *Adaptive background mixture models for real-time tracking*, [online], Dostupné na: [http://www.ai.mit.edu/projects/vsam/Publications/stauffer\\_cvpr98\\_track.pdf](http://www.ai.mit.edu/projects/vsam/Publications/stauffer_cvpr98_track.pdf) citováno 3.5. 2012
- [10] P. Wayne Power Johann, Schoonees, W., *Understanding Background Mixture Models for Foreground Segmentation*, [online], Dostupné na: <http://www.cse.psu.edu/~rcollins/CSE586Spring2010/papers/emBGsubtractAboutSandG.pdf>, citováno 3.5.2012
- [11] Thul, E., *Adaptive Background Mixture Models for Real-Time Tracking*, [online], Dostupné na: <http://www.cs.mcgill.ca/~ethul/pub/course/ecse626/project-pres.pdf> citováno 3.5. 2012
- [12] Tomáš Jelínek: *Detekce pohybujících se objektů ve video sekvenci*. Brno, 2007, diplomová práce, FIT VUT v Brně. Dostupné na: [http://www.fit.vutbr.cz/study/DP/DP.php?id=5197&y=\\*&t=Detekce+pohybuj%EDc%EDch](http://www.fit.vutbr.cz/study/DP/DP.php?id=5197&y=*&t=Detekce+pohybuj%EDc%EDch), citováno 3.5. 2012
- [13] Petrovský, P., *Barevné modely*, [online], Dostupné na: <http://www.uamt.feec.vutbr.cz/vision/TEACHING/MPOV/mpov.html.en> citováno 3.5. 2012
- [14] Fatura, K., *Eliminace stínů pohybujících se objektů*, [online], Praha 24.11.2006, Dostupné na: [karel.aicko.cz/kaja2/files/stiny.pdf](http://karel.aicko.cz/kaja2/files/stiny.pdf), citováno 3.5.2012
- [15] Petr Hýna: *Detekce rohů v obraze*, bakalářská práce, Brno, FIT VUT v Brně, 2007, [www.fit.vutbr.cz/study/DP/rpfile.php?id=4930](http://www.fit.vutbr.cz/study/DP/rpfile.php?id=4930), citováno 4.5.2012
- [16] Horák, K., *Lokální příznaky a korespondence*, [online], Dostupné na: [http://midas.uamt.feec.vutbr.cz/POV/Overview/overview\\_cz.php](http://midas.uamt.feec.vutbr.cz/POV/Overview/overview_cz.php), citováno 4.5.2012
- [17] Svoboda, T., *I. cvičení: Matlab; Jasové transformace*, ukázkový kód, [online], Dostupné na: [http://cmp.felk.cvut.cz/cmp/courses/ZSO/1/data\\_basics/data\\_basics.m](http://cmp.felk.cvut.cz/cmp/courses/ZSO/1/data_basics/data_basics.m) citováno 4.5.2012



- [18] Elgammal, A., Harwood, D., Davis, D., *Non-parametric Model for Background Subtraction*, [online], Dostupné na: <http://www.cs.umd.edu/users/elgammal/docs/eccv2000.pdf> citováno 4.5.2012
- [19] Wesolkowski S., *Color Image Edge Detection and Segmentation*, [online], Waterloo, Ontario, Canada, 1999, © Slawomir B. Wesolkowski 1999, Dostupné na: <http://www.collectionscanada.gc.ca/obj/s4/f2/dsk3/OWTU/TC-OWTU-105.pdf> citováno 18.5.2013
- [20] Jan, J., *Fúze obrazů*, Brno, 2011, © J. Jan 2011
- [21] Horák, K., *Lokální příznaky a korespondence* [online], Brno, © Karel Horak 2010, Dostupné na [http://midas.uamt.feec.vutbr.cz/POV/pov\\_cz.php](http://midas.uamt.feec.vutbr.cz/POV/pov_cz.php), citováno 18.5.2013
- [22] Horák, K., *Dynamické obrazy* [online], Brno, © Karel Horak 2010, Dostupné na [http://midas.uamt.feec.vutbr.cz/POV/pov\\_cz.php](http://midas.uamt.feec.vutbr.cz/POV/pov_cz.php), citováno 18.5.2013
- [23] J. Jan, *Diskrétní 2D lineární transformace*, Brno, 2011, © J. Jan 2011
- [24] Zitová, B., Flusser, J. *Image registration methods: a survey*, [online], Image Vision Comput. 21(11): 977-1000 (2003). Dostupné na: <https://www.sparkfun.com/products/9694>, citováno 4.1.2013
- [26] Petr Hýna: *Detekce rohů v obraze, bakalářská práce*, Brno, FIT VUT v Brně, 2007, [online] [www.fit.vutbr.cz/study/DP/rpfile.php?id=4930](http://www.fit.vutbr.cz/study/DP/rpfile.php?id=4930), citováno 4.5.2012
- [27] *Conner detection*, [online], last modified on 28 April 2013, Dostupné na: [http://en.wikipedia.org/wiki/Corner\\_detection](http://en.wikipedia.org/wiki/Corner_detection), citováno 4.1.2013
- [28] P. Dias, *Haralick Corner Detektor*, [online], 05/11/2003 IEETA/Universidade de Aveiro, 05/11/2003, Dostupné na: [http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/DIAS2/](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/DIAS2/), citováno 4.1.2013
- [29] DUŠEK, S. *Určení parametrů pohybu ze snímků kamery*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 76 s. Vedoucí diplomové práce Ing. Ilona Kalová, Ph.D. [online], Dostupné na:

[www.vutbr.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=15400](http://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=15400)

citováno 4.1.2013

[30] *Hledání korespondencí*, CVUT, [online], Dostupné na:

[http://cw.felk.cvut.cz/doku.php/courses/a4m33mpv/cviceni/2\\_hledani\\_korespondenci/start](http://cw.felk.cvut.cz/doku.php/courses/a4m33mpv/cviceni/2_hledani_korespondenci/start),

citováno 4.1.2013

[31] Horák, K., *Porozumění obsahu obrazu*, [online], Brno, © Karel Horak 2010, Dostupné na

[http://midas.uamt.feec.vutbr.cz/POV/pov\\_cz.php](http://midas.uamt.feec.vutbr.cz/POV/pov_cz.php), citováno 18.5.2013

[32] Perďoch, M., extrema.\* J. Matas a M. Perďoch, *Hledání korespondencí I, detekce bodů zájmu*, [online], Dostupné na:

[http://cw.felk.cvut.cz/doku.php/courses/a4m33mpv/cviceni/2\\_hledani\\_korespondenci/start](http://cw.felk.cvut.cz/doku.php/courses/a4m33mpv/cviceni/2_hledani_korespondenci/start)

citováno 4.1.2013

[33] Lewis, J., P., *Fast normalized X-correlation*, [online], Dostupné na

<http://scribblethink.org/Work/nvisionInterface/nip.html>, citováno 18.5.2013

[34] Mikolajczyk, K., Schmidt, C., *Scale & Affine Invariant Interest Point Detectors* [online], International Journal of Computer Vision 60(1), 63–86, 2004, ©2004, Dostupné na:

<http://www.cse.unr.edu/~bebis/CS773C/ObjectRecognition/Papers/Mikolajczyk04a.pdf> ,

citováno 18.5.2013

[35] Seki, M., Wada, T., Fujiwaray, H., Sumi, K., *Background Subtraction based on Cooccurrence of Image Variations*, [online], Computer Vision and Pattern Recognition, 2003. IEEE, Dostupné na:

[nguyendangbinh.org/Proceedings/CVPR/2003/pdf/papers/P02\\_33.pdf](http://nguyendangbinh.org/Proceedings/CVPR/2003/pdf/papers/P02_33.pdf)

citováno 18.5.2013

[36] Havlíková, M., *Digitální analýza a zpracování signálů*, MZPD-2012/13, Tutoriál č.8 , část Kepsrální analýza, 2013, Brno, citováno 18.5.2013

[37] *Sledování objektů*, CVUT, [online], Dostupné na:

[http://cw.felk.cvut.cz/doku.php/courses/a4m33mpv/cviceni/4\\_sledovani\\_objektu/start](http://cw.felk.cvut.cz/doku.php/courses/a4m33mpv/cviceni/4_sledovani_objektu/start) ,

citováno 18.5.2013

# Seznam příloh

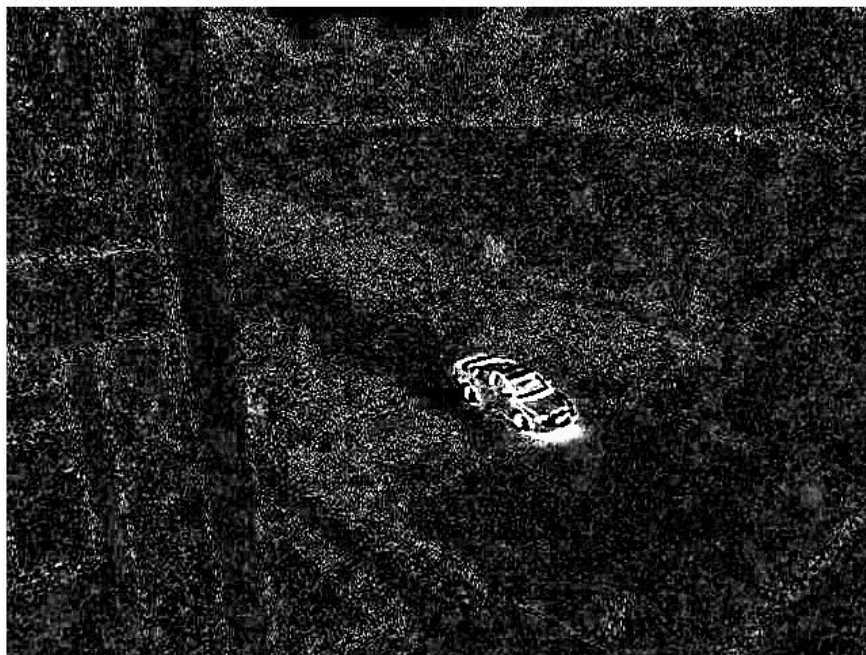
Příloha 1. Ukázkové obrázky

Příloha 2. Blokové diagramy vybraných programů

Příloha 3. CD s ostatními přílohami

1. Zdrojové kódy
2. Popis k vytvořeným scénám
3. Testovací videa různých scén
4. Ukázkové obrázky a videa

Rozdílový obrázek bez filtrace



**Příloha 1. Obr. 1 Rozdílový obrázek Basic Background Subtraction Method**

Aktualni obrazek



Příloha 1. Obr. 2 Running Gaussian average Scéna dálnice Iakt

Pohyb v obrazku podle vzdl. od stredni hodnoty



Příloha 1. Obr. 3 Running Gaussian average Scéna dálnice Popředí

Pozadí obrazku



Příloha 1. Obr. 4 Running Gaussian average Scéna dálnice Pozadí

Aktualni obrazek



Příloha 1. Obr. 5 Running Gaussian average Scéna Křižovatka2 Iakt

Odectene popredi



Příloha 1. Obr. 6 Running Gaussian average Scéna Křižovatka2 Popředí

Pozadí obrazku



Příloha 1. Obr. 7 Running Gaussian average Scéna Křižovatka2 Pozadí

Aktualni obrazek



Příloha 1. Obr. 8 Running Gaussian average Scéna dálnice2 Iakt

Odectene popredi



Příloha 1. Obr. 9 Running Gaussian average Scéna dálnice2 Popředí

Pozadí obrázku



**Příloha 1. Obr. 10 Running Gaussian average Scéna dálnice2 Popředí**

Aktuální obrázek



Pozadí obrázku



**Příloha 1. Obr. 11 Temporar median filter Scéna dálnice**





**Příloha 1. Obr. 12 Temporar median filter Scéna dálnice2**

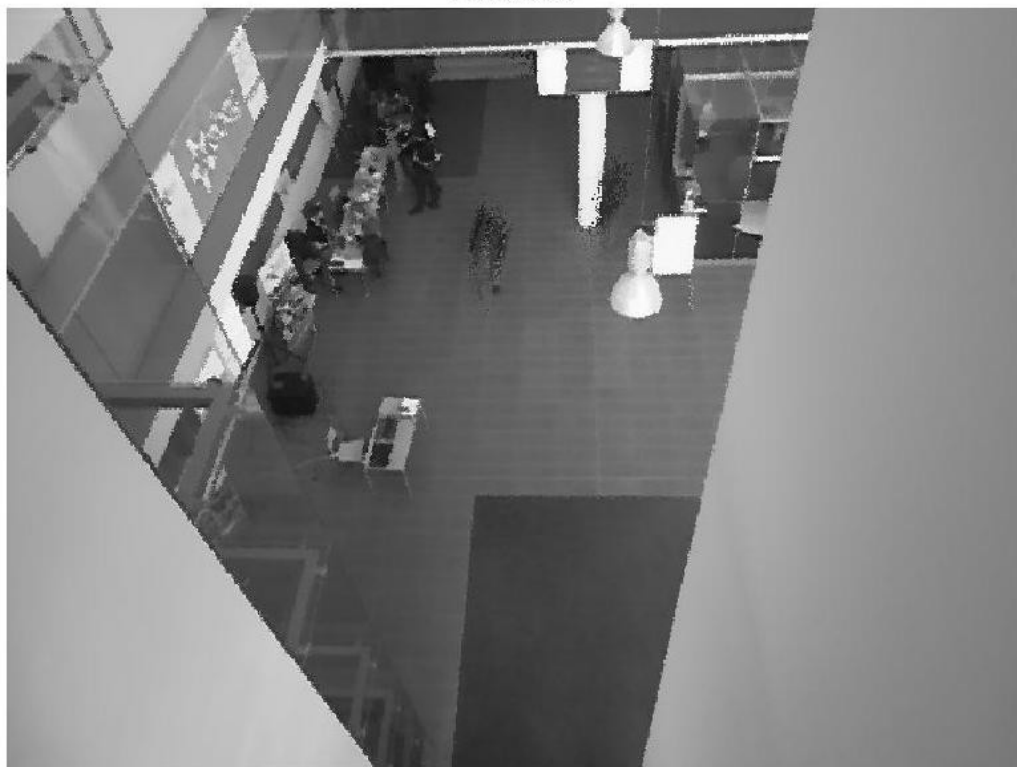


**Příloha 1. Obr. 13 Temporar median filter Scéna křižovatka (auto uprostřed stojí)**



**Příloha 1. Obr. 14 Temporar median filter Scéna křižovatka krátký buffer (auto uprostřed stojí)**

Pozadí obrázku



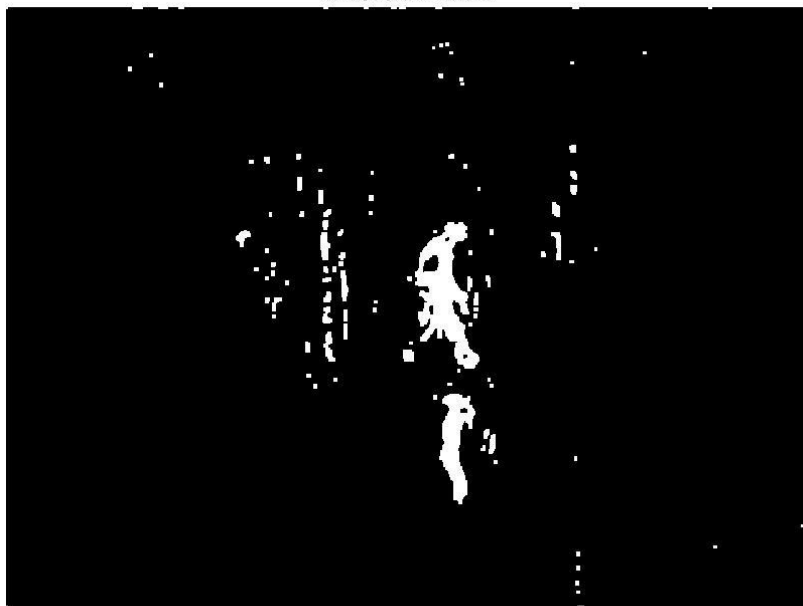
**Příloha 1. Obr. 15 Temporar mode filter**

Aktualní obrazek



**Příloha 1. Obr. 16 MoG Scéna chodba Iakt bez stabilizace**

Morfologicky upravené popředí



**Příloha 1. Obr. 17 MoG Scéna chodba Popředí bez stabilizace**

Pozadí obrázku70



**Příloha 1. Obr. 18 MoG Scéna chodba Pozadí bez stabilizace**

Obrazek s 117 významnými body způsob 1



Obrazek s 113 významnými body způsob 2



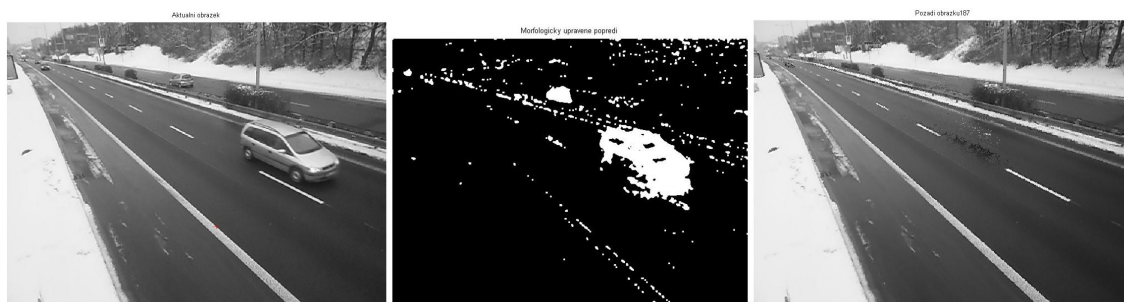
Obrazek s 83 významnými body způsob 3



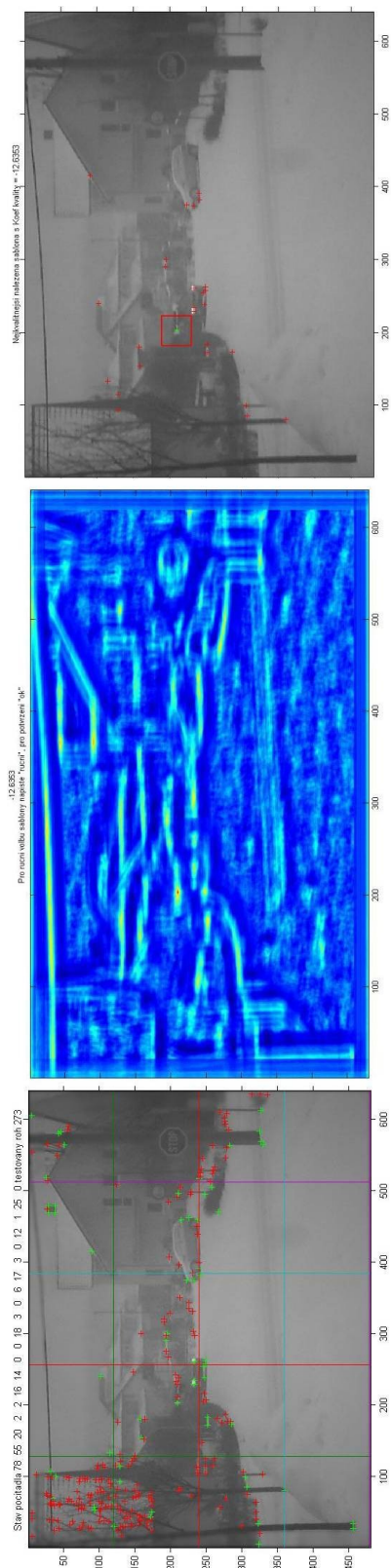
Obrazek s 190 významnými body způsob 4



**Příloha 1. Obr. 19 Harrisův operátor způsob (modifikace) 1-4**

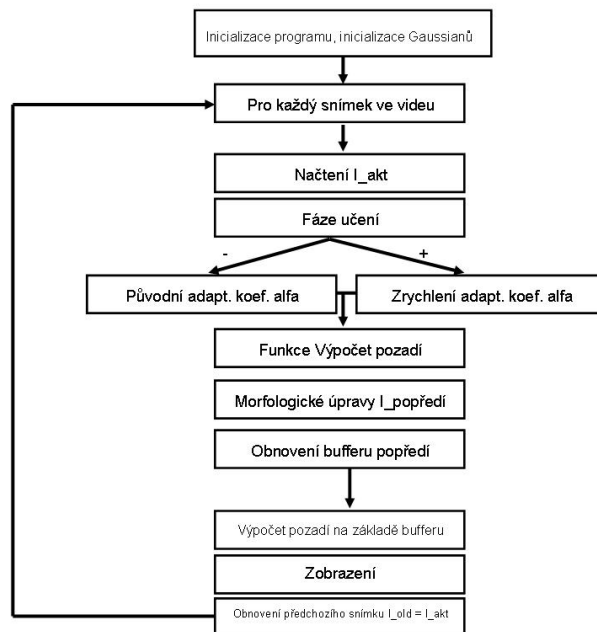


**Příloha 1. Obr. 20 MoG Scéna dálnice3**



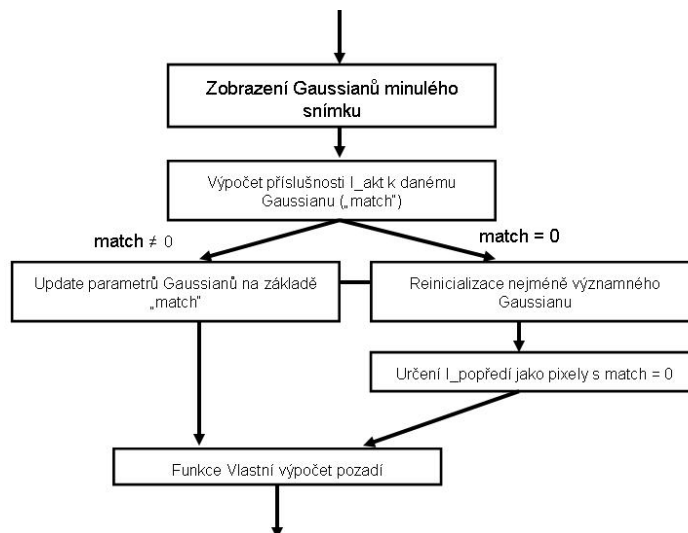
Příloha 1. Obr. 21 Automatický výběr šablony, silné sněžení, zoom

# Mixture of Gaussian



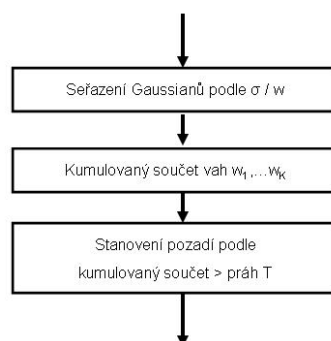
Příloha 2. Obr. 1 Blokový diagram algoritmu MoG

## Funkce Výpočet pozadí



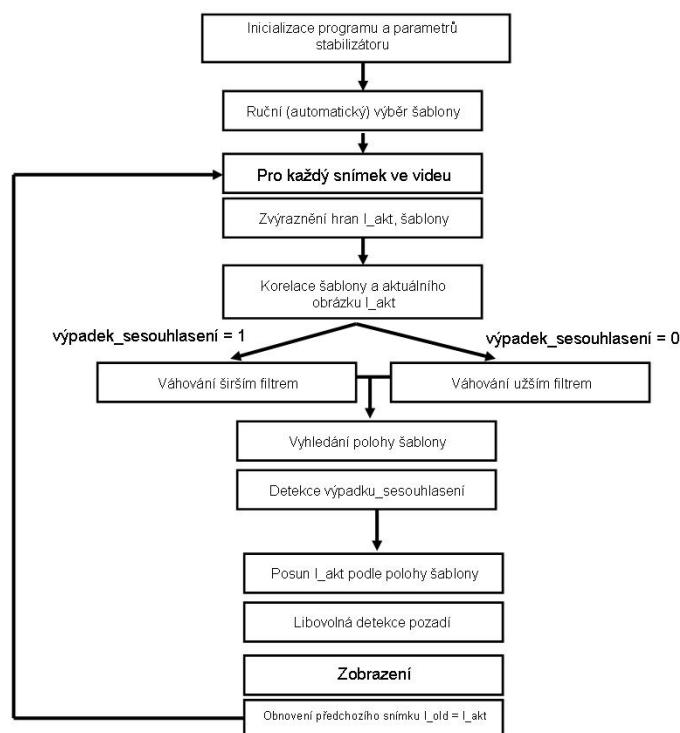
Příloha 2. Obr. 2 Funkce použitá v algoritmu MoG

## Funkce Vlastní výpočet pozadí



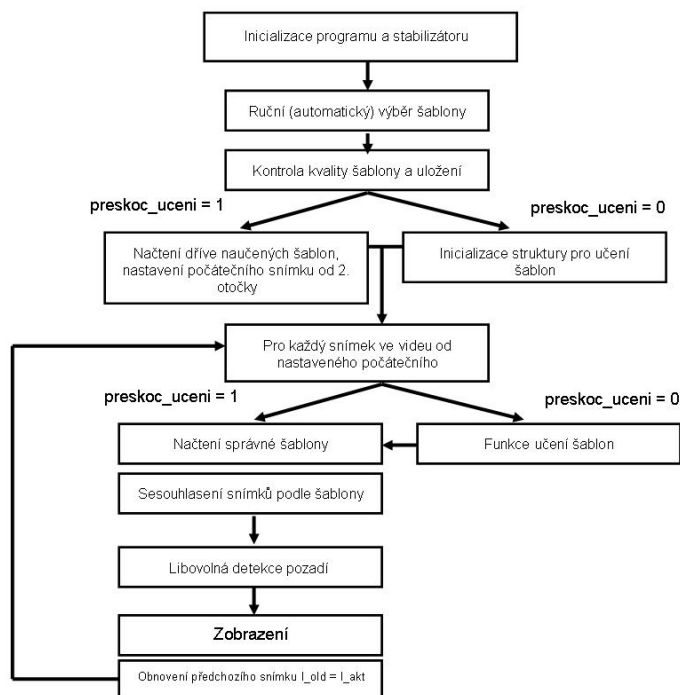
Příloha 2. Obr. 3 Funkce použitá ve funkci Výpočet pozadí

## Statická kamera se stabilizátorem



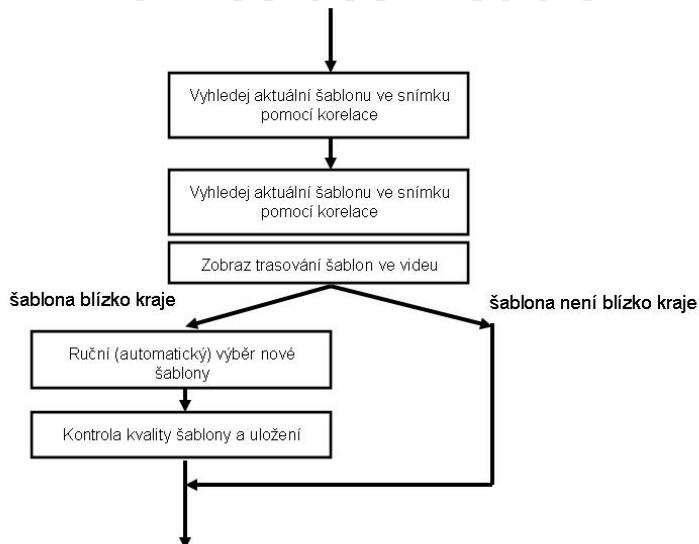
Příloha 2. Obr. 4 Statická kamera s optickým stabilizátorem na základě korelace

# Otočná kamera



Příloha 2. Obr. 5 Otočná kamera

# Funkce učení šablon



Příloha 2. Obr. 6 Funkce použitá v programu Otočná kamera